FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

A Comparative Perceptual Study of Soft Shadow Algorithms

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Michael Hecher
Matrikelnummer 0625134

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Dipl.-Ing. Matthias Bernhard
Dipl.-Ing. Dr.techn. Oliver Mattausch

Wien, 06.09.2012

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

A Comparative Perceptual Study of Soft Shadow Algorithms

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Visual Computing

by

Michael Hecher
Registration Number 0625134

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Assistance: Dipl.-Ing. Matthias Bernhard
Dipl.-Ing. Dr.techn. Oliver Mattausch

Vienna, 06.09.2012

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Michael Hecher
Steinfeldstr. 42, 2731 St. Egyden am Stfd.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Acknowledgements

I would like to thank my supervisors Matthias Bernhard, Oliver Mattasch, Daniel Scherzer, and Michael Wimmer for their help on this thesis. I would also like to thank Michael Schwarz for allowing us to use his implementation of the Backprojection algorithm. And of course my family for their support.

Abstract

While a huge body of soft shadow algorithms has been proposed, there has been no methodical study for comparing different real-time shadowing algorithms with respect to their plausibility and visual appearance. Therefore, a study was designed to identify and evaluate scene properties with respect to their relevance to shadow quality perception. Since there are so many factors that might influence perception of soft shadows (e.g., complexity of objects, movement, and textures), the study was designed and executed in a way on which future work can build on. The novel evaluation concept not only captures the predominant case of an untrained user experiencing shadows *without* comparing them to a reference solution, but also the cases of trained and experienced users. We achieve this by reusing the knowledge users gain during the study. Moreover, we thought that the common approach of a two-option forced-choice-study can be frustrating for participants when both choices are so similar that people think they are the same. To tackle this problem a neutral option was provided. For time-consuming studies, where frustrated participants tend to arbitrary choices, this is a useful concept. Speaking with participants after the study and evaluating the results, supports our choice for a third option. The results are helpful to guide the design of future shadow algorithms and allow researchers to evaluate algorithms more effectively. They also allow developers to make better performance versus quality decisions for their applications. One important result of this study is that we can scientifically verify that, without comparison to a reference solution, the human perception is relatively indifferent to a correct soft shadow. Hence, a simple but robust soft shadow algorithm is the better choice in real-world situations. Another finding is that approximating contact hardening in soft shadows is sufficient for the "average" user and not significantly worse for experts.

Kurzfassung

Obwohl diverse Algorithmen zur Echtzeitsimulation von “weichen” Schatten existieren, gibt es bis jetzt keine methodische Studie, welche verschiedene dieser Algorithmen in Bezug auf ihre Plausibilität und Qualität hin untersucht. Deshalb wurde im Zuge dieser Arbeit eine Studie entworfen, mit der auf systematische Weise Eigenschaften identifiziert werden sollen, die für die wahrgenommene Qualität von weichen Schatten in virtuellen Szenen relevant sind. Da es viele Faktoren gibt, welche die Wahrnehmung von weichen Schatten beeinträchtigen könnten (z.B. Komplexität, Bewegung und Texturen von Objekten bzw. die Komplexität der Schatten), wurde eine Studie konzipiert und durchgeführt, auf deren Basis zukünftige Arbeit aufbauen können. Das neuartige Evaluationskonzept der Studie erfasst dabei nicht nur den vorherrschenden Fall des unerfahrenen Benutzers, sondern auch jene von geschulten und erfahrenen Anwendern. Dadurch gehen die erworbenen Kenntnisse, welche Teilnehmer durch die Studie gewonnen haben, nicht verloren und können zur Auswertung anderer Erfahrungsgrade genutzt werden. Im Gegensatz zu früheren Studien hatten Teilnehmer statt zwei, drei Antworten zur Auswahl. Zusätzlich zu den Möglichkeiten “links besser” bzw. “rechts besser” wurde die Option “sind gleich” angeboten. Dies ist nützlich, wenn “links” und “rechts” so ähnlich sind, dass sie der Benutzer als ident empfindet. Solche Fälle erzeugen Rauschen, da sich Teilnehmer willkürlich entscheiden müssen. Des Weiteren sind solche Situationen frustrierend, wenn nicht die dritte Wahlmöglichkeit zur Verfügung steht. Die Ergebnisse der Studie sind nützlich, um zukünftige Schattenalgorithmen zu entwickeln und um diese besser evaluieren zu können. Darüber hinaus wird es Entwicklern ermöglicht, bessere Kosten-Nutzen-Entscheidungen für ihre Anwendungen zu treffen. Ein wichtiges Ergebnis der Studie ist, dass, wenn kein Vergleich mit einer Referenzlösung zur Verfügung steht, die menschliche Wahrnehmung relativ unempfindlich in Bezug auf die Plausibilität weicher Schatten ist. Daher sind in praktischen Anwendungsfällen einfache, aber robuste Algorithmen, plausiblen, aber fehleranfälligen Algorithmen vorzuziehen.

Contents

CHAPTER 1

Introduction

Shadows are an important cue for the perception of spatial relationships between objects. A sample that visualizes this fact is shown in Figure 1.1. Without shadows it is not clear where the objects are located. However, this is not the only reason why shadows are used in computer generated images. They are also an aesthetic tool that adds plausibility to virtual environments. Through the years, the increasing computational power of computers has allowed more and more sophisticated solutions to compute shadows. One noticeable change for consumers was the introduction of *soft shadows*, which replaced *hard shadows* in computer games. Hard shadows are cast by point lights, which have no extent and can only be approximated in real-life (e.g., by flashlight). The more common case of area light sources (e.g. light bulbs, or neon lights) produces soft shadows (see Figure 1.2).

Because of this real-time algorithms for soft shadows have been an important branch of real-time rendering research for many years. The problem, however, is a complex one. An area

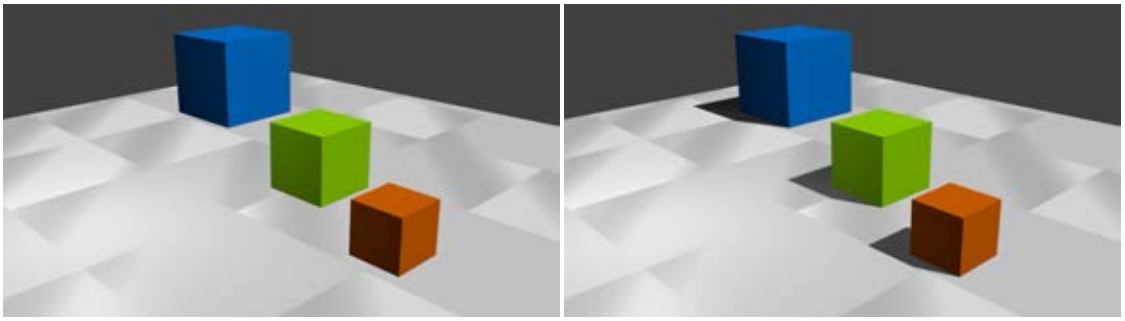

Figure 1.1: The following sample is often given to emphasize the importance of shadows for spatial perception: In the left image it is not clear whether the blue and green cube are floating in the air or if they are placed on the ground. In the right image the spatial relation between the objects becomes clear through the presence of shadows.

light source can be seen has a collection of an infinite amount of point lights. The sum of all point lights generates the soft shadow. In practice an infinite amount of point lights cannot be processed, and even approximations with a finite amount are not practical, as hundreds of point lights are needed to produce good results. The goal is to find clever ways to provide a *plausible* approximation of the appearance of physical soft shadows within a time budget of 60 frames per second. Quality per time budget and robustness of shadow techniques are consequently of great concern for developers of real-time applications.

1.1 Research Questions

In this work we want to solve the following research questions:

> *How plausible do soft shadows have to be?*
>
> *How can different degrees of user experiences be captured in a study?*

The first question is important for developers and researchers to increase performance in real-time applications and to create faster and more plausible algorithms. But to answer this question, different levels of user experience have to be considered that range from inexperienced to experienced users. Hence we have to design a user study that can handle different levels of user experiences.

1.2 Contributions

By solving the research questions above, we make the following contributions to the research community:

> *Through an experiment we show that approximating contact hardening in soft shadows is sufficient for the "average" user and not significantly worse for experts.*
>
> *The novel methodology allows us to capture and evaluate different degrees of user experiences.*

Besides handling multiple levels of experiences in an easy manner (without training people beforehand), the methodology is designed to suite the participants' way of making decisions. Since the design allows participants to make neutral choices, we can pursue the reason for these decisions as we will see in Chapter 6.

1.3 Overview

The goal of this work is to find and investigate factors which influence the perceived quality of a shadow algorithm. We will therefore give an overview of similar studies that investigate the perception of realism or the quality of algorithms.

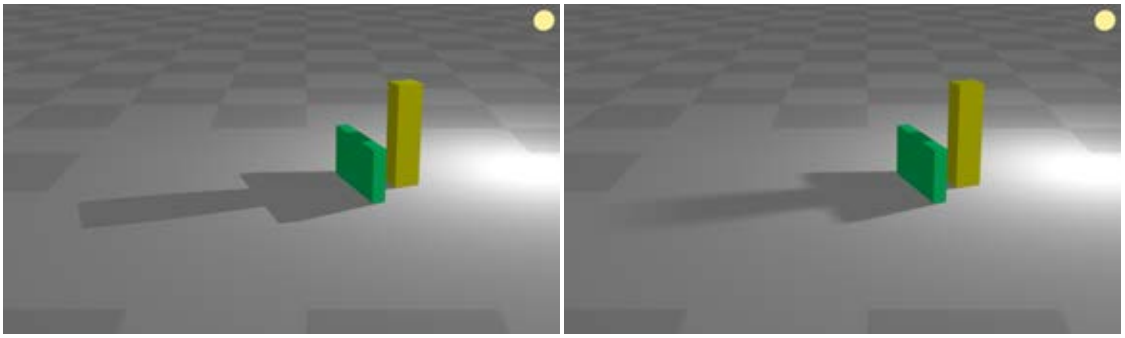

Figure 1.2: The left image shows a scene where a hard shadow algorithm was used to generate the shadow. In the right image a soft shadow algorithm was used, whereas in the right image the shadow gets softer the farther away it is from the object. Which is why such shadows are known as *soft shadows*.

We will then discus the current research results in the area of soft shadow algorithms and take a look at the advantages and disadvantages of each of these algorithms. Extensive research has produced a wide variety of algorithms, so the focus will lie on a few well known techniques. This will give an overview of the current state-of-the-art in soft shadow simulation and allows us to select representative algorithms for the study.

Next we discuss the design of the user study in detail, e.g., why a subset of four particular algorithms was selected and which factors influenced the design decisions for the experiment.

Because of the sophisticated design, the statistical evaluation is complicated and we will discuss the mathematical background in great detail.

Finally we investigate the outcome of the study and draw conclusions from the results of the experiment and discuss possible future work in the field of soft shadow perception.

CHAPTER 2

Psychophysical Studies

In this chapter we will take a look at psychophysical studies that are similar to ours and investigate the perception of realism or the quality of algorithms. Psychophysical research was introduced by Fechner in 1869 in his work *"Elemente der Psychophysik"* [18] which is based on preliminary work of Ernst Heinrich Weber. In general psychophysic is described as "the scientific study of the relation between stimulus and sensation" [22]. There are several psychophysical methods to quantize human perception [20, 22].

For our psychophysical study we employed the method of *pairwise comparison*. The principal idea of this method is to let participants compare two stimuli (e.g. images) to each other and let them select one based on some criteria. There are different ways to evaluate the outcome of such experiments. In the first section of this chapter we will discuss multiple approaches on which our experiment is funded.

Afterwards we will discuss the findings of other studies to see how this work fits into the latest shadow perception research. Investigations in this area show that our work is complementary to others as there has not been any methodical study for comparing different real-time soft shadow algorithms with respect to their plausibility and visual appearance.

2.1 Approaches

We provide a study about the perception of distinct classes of soft shadow algorithms and evaluate scene properties with respect to their relevance to soft shadow perception. Similar psychophysical experiments by Yu et al. were executed in the area of *indirect illumination* [55]. They suspected that the accurate computation of visibility between two points is perceptually not necessary to generate plausible *global illumination* effects. They were able to validate this assumption by designing two studies based on *paired comparison plus category* [41] and *ordinal rank order* [5]. Paired comparison with categories is equal to paired comparison except that instead of choosing between two stimuli, subjects have to select from categories (e.g. not similar, slightly similar, moderately similar, very much similar, and extremely similar). Yu et

Figure 2.1: Sattler et al. could show that in case of the Sanford Bunny model 1% of the original number of triangles was sufficient to produce realistic soft shadows for 90% of the tested persons. (Source [40].)

al. used this approach to compute the similarity of approximate visibility algorithms for *indirect illumination*. As stimuli videos showing the output of these algorithms were used. Ordinal rank order was employed to evaluate perceived realism. For this method participants had to order the videos in terms of perceived realism. The problem with these approaches is that categories like "very much similar" and "extremely similar" do not necessarily scale equally between subjects or have the same meaning to them. A linear scale is preferable. Computing the perceived realism with a *ordinal rank order* experiment is a good idea, but it requires additional participants and time.

To save time Rubinstein et al. designed a large scale user study that employed a *linked-paired comparison design* [14] to evaluated image-retargeting methods [39]. The aim of the study was to find out if computational image distance metrics can be used to predict human retargeting perception. Participants had to compare retargeted images and select the better looking one. Because the number of comparisons for each participant would have been too high for 8 algorithms and 37 images, they had to reduce the sample set. With *linked-paired comparison* the number of comparisons could be reduced. Subjects only had to vote 12 instead of 28 times for each image. The results of the study show that image features previously not used for a metric achieved better results. They also indicate that some methods are consistently better than others and that some qualities in images are more important to observers.

The work of Rubinstein et al. is related to studies by Gutierrez et al. [24] and Ledda et al. [31] in terms of processing pairwise comparison data and presentation of the results. Gutierrez et al. let users compare *tone mapping operators* that were applied to *high dynamic range* (HDR) images. Such operators are used to displayed HDR images on non HDR displays. In their first experiment HDR images were shown to participants on two LCD monitors. For each monitor a different tone mapping operator was used. Between them a HDR display showed the image without applied operators. Subjects then had to decide which operator produced better results compared to the reference image on the HDR display. This approach is useful, when the performance of algorithmic processing has to be tested against the optimal solution. This provides a worst case scenario where participants know exactly what a stimulus should look like. The outcome of the study was visualized by assigning ranks (1: best, 2: second best, ...) to operators for each image, depending on the number of votes it received. The ranks were then used to group operators with comparable performance. The problem with this approach is that by assigning ranks of positive integers, the real proportion of how many persons voted for which algorithm is lost. So we only know which algorithms are comparable and which perform better,

but we cannot say how much better.

The research aims of Gutierrez et al., Ledda et al., and Rubinstein et al. are similar to ours. We therefore decided to adapt the design of these studies. At the same time we wanted to improve them in areas, where we think informative values are lost, user consistency suffers, or better evaluation methods are available. For instance ranks can be computed with the Bradley-Terry-Luce model which lets us see how much better a method is over another [8].

2.2 Shadow Studies

We will investigate how specific scene and soft shadow properties affect the perceived plausibility of shadows. Such psychophysical evaluation of quality and plausibility, has not received much attention in computer sciences. There are few publications that evaluate quality in computer generated images or try to find out how plausible specific features in scenes, like materials or shadows, have to be for users. Most research concentrates on answering questions like "Which aspects contribute to the quality of an image?" or "Do users prefer some technique over a simpler one?". Other works focus on finding metrics to approximate shadows reflections, materials, and other features in images for regions where users are insensitive to errors.

Boulenguez et al. defined a metric to quantify the quality of computer generated images [7]. This was done by an experiment where participants had to assign scores to the overall quality of several images. They then had to rate the aspects of this quality. The outcome of the study was that aspects like accurate simulation, good contrast, and absence of noise, were more important than precise anti-aliasing and faithful color bleeding.

Mania et al. designed an experiment to explore if improved rendering quality had any impact on subjective impressions of illumination and perceived presence in a virtual environment [33]. Different levels of shadow accuracy were used to influence rendering quality. After watching a virtual scene through a Head Mounted Display, participants received two questionnaires. One about the perceived presents in the virtual environment and a second one investigating subjective impressions of lighting (e.g. warm, comfortable, spacious). The results show that there is a positive correlation between presence and impressions of lighting and increasing render quality in virtual environments. This means that soft shadows have to be preferred over hard shadows if users should feel immersed into a virtual environment. Or in other words: To make a virtual world believable, shadows have to be plausible.

Raya et al. compared different shadow and reflection techniques in terms of quality [37]. The authors wanted to find out if users prefer a specific kind of simulation (e.g., soft over hard shadows, or physically correct soft shadows over estimated soft shadows). The outcome of the experiment was that users do not prefer images that are generated with more sophisticated techniques. Inexperienced users are not able to determine inconsistencies between different kinds of shadow techniques. This indicates that soft shadows do not significantly improve quality. So for situations where the image quality should stay roughly the same on different devices, hard shadows can be used on weaker hardware without significantly decreasing quality (e.g. PC vs. mobile phone). Note that this does not contradict the findings of Mania et al. Though soft shadows do not significantly improve quality, they increase believability of virtual worlds. So if we

want to give users the impression of presents at the lowest possible cost, we have to investigate the perception of soft shadows.

There has been research on shadow perception by psychological studies of Wanger et al. They could prove that shadows are a major factor in the spatial perception of objects [51]. They used three experiments in which participants had to position, rotate, and scale objects. In each of the three experiments different cues were tested (hard shadows, textures, perspective, motion, and elevation) to investigate which ones effect *spatial perception*. The results show that shadows offer the most significant cues. In following experiments Wanger et al. found out that soft shadows do not improve spatial perception. Quite to the contrary, they make it more difficult for participants to recognize the object by its soft shadow [50]. So in reverse we can assume that it is also hard for people to recognize the correct soft shadow of an object. If this is a valid assumption, users should have difficulties distinguishing correct soft shadows from faked ones. Of course this needs to be proven.

All this knowledge is used to exploit human perception for accelerating soft shadow rendering in regions where the sensitivity is low and hence a lower quality shadow is sufficient [40, 44, 49]. Sattler et al. observed that strongly simplified versions of three dimensional shapes are often enough to provide a plausible impression of a shadow [40] (see Figure 2.1). This is another indication that our assumption is correct. There are other works by Vangorp et al. [49] and Schwarz et al. [44] that are also complementary to ours. Like Sattler et al. they address the question of scaling particular types of algorithms. There has, however, not been any methodical study for comparing different real-time soft shadow algorithms with respect to their plausibility and visual appearance.

CHAPTER 3

Real-Time Soft Shadow Algorithms

This chapter discus the current research results in the area of soft shadow algorithms. The basis for most of these algorithms provided Crow and Williams, who introduced *shadow volumes* [13] and *shadow mapping* [52] respectively. These publications are well-known in the field of computer graphics. They have been improved by other researchers in the last decades in terms of performance, reduced aliasing artifacts, and robustness [32, 47, 53]. Since graphics hardware has become powerful enough to simulate shadows in real-time, the number of applications using shadows has increased significantly.

Though hard shadow algorithms make a scene look realistic, soft shadows provide even more plausibility by allowing the simulation of area light sources which are predominant in real life. Early implementations of soft shadows were expensively pre-computed and stored in maps in order to achieve real-time performance at runtime. Famous techniques are *light maps* and *photon maps* [27]. These methods are used for static scenes, because in dynamic scenes changing light conditions require constantly recomputing these maps. In recent years scenes have become more dynamic and applications simulate day and night. These developments require real-time soft shadow algorithms. Research in this area includes the correct estimation of the hard (*umbra*) and the soft parts (*penumbra*) of the shadow (see Figure 3.1).

As all presented soft shadow algorithms are based on shadow mapping or shadow volumes, we will take a look at implementation details of these techniques. Afterwards we will discuss how they can be adopted to generate soft shadows. In the end we will compare them to each other. Based on this comparison we can select soft shadow algorithms for the study.

3.1 Real-Time Hard Shadows

As mentioned above, most of the known soft shadow algorithms are based on shadow mapping or shadow volumes. These algorithms are intended to create “hard” shadows. Hard shadows represent shadows of a point light source and have no soft edges. Light bulbs and other common light sources, however, do not produce hard shadows. Because the light is emitted over an

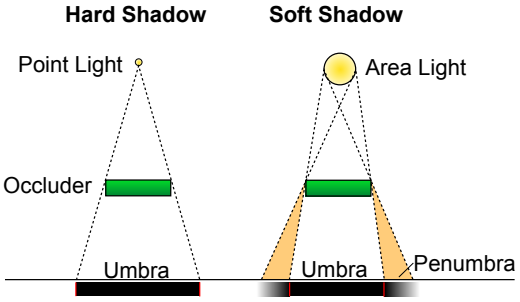

Figure 3.1: For the point light on the left there can only be an umbra, because a point cannot be partly occluded. For the area light on the right there are places in the shadow where the light is partly visible, called the penumbra (marked orange).

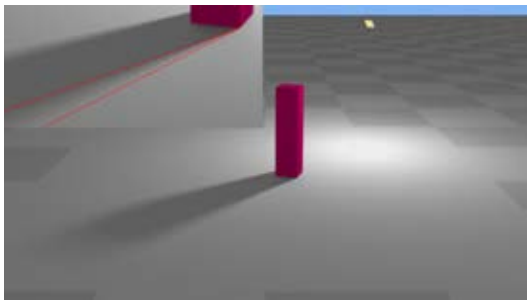

Figure 3.2: As the red lines indicate the term contact shadows refers to situations where shadows are sharp near the object and get softer when they are farther away.

area, penumbras become visible (see Figure 3.3). The size of the penumbra increases with the distance between the shadow caster and the shadow receiver. On the other hand shadows become harder if the distance decreases. This effect is called *contact hardening* (see Figure 3.2). There have also been other proposals to generate shadows. For instance the occluder geometry can be projected onto a plane and used as a shadow (*projected shadows*) [6]. The problem with this proposal is that it only works for planar scenes and does not handle self-shadowing. In current state-of-the-art graphics applications, worlds are complex and, for instance, contain mountains, water, architecture, and other objects. Planar surfaces are rare in such scenes, which is why this algorithm is not used in games and other interactive applications. All other algorithms in this work are based on shadow mapping or shadow volumes, so we will take a look at how those two algorithms work, before their various soft shadow implementations are discussed in the next section.

Shadow Volumes

Shadow volumes were introduced by Franklin Crow [13] and were extended to take advantage of stencil buffers on graphics hardware by Tim Heidemann. At first shadow volumes for each light source have to be created. This is done by finding all *silhouette edges* of an occluder from the light point of view. The edges are then extended in the direction away from the light source. So for each edge, a so called *shadow quad* is created. All shadow quads together form the shadow volume of an object. For infinite light sources the volume is extended to infinity. For finite lights it is extended as far as the light reaches. So for each silhouette edge a new polygon has to be generated for each light source. On graphics hardware that support geometry shaders the generation of shadow volumes can be simplified by determining the silhouette of the shadow caster in geometry stage of the rendering pipeline.

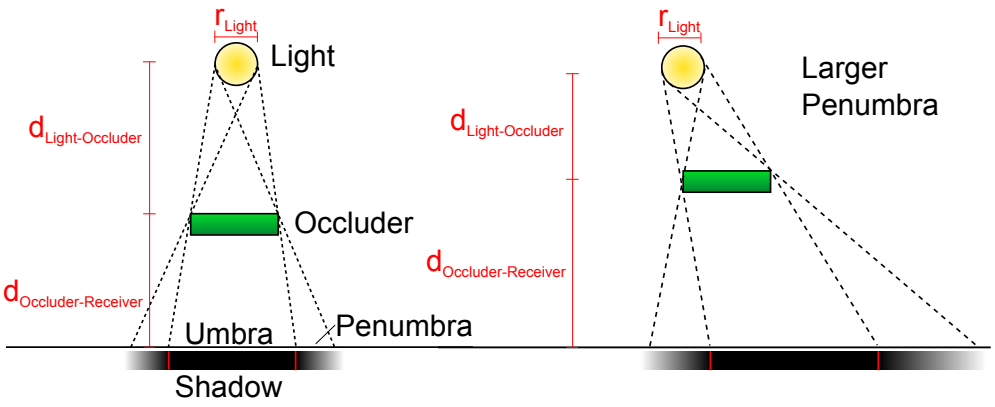

Figure 3.3: The spatial relationship of the light source, occluder, and shadow receiver influence the size of the penumbra.

After the volume has been generated different approaches can be used to determine whether a pixel lies within the shadow volume. As all of them utilize the stencil buffer in some way, we will only discuss one approach purposed by Heidemann, which is call the *depth pass* test. First the front-faces of the shadow volumes are rendered into the stencil buffer, which increments the buffer for each triangle inside the screen. Then the back-faces are rendered and the stencil buffer is decremented for each triangle. Now all pixels that have a stencil value of 0 are lit. All others lie inside a shadow volume and inside the umbra (see Figure 3.4).

The advantages of this method are the pixel-exact generation of hard shadows, the simple implementation on graphics hardware that supports geometry shaders, and the simulation of point lights. Disadvantages lie in special cases that have to be handled differently. If the observer's position is inside a shadow volume, the stencil buffer has to be initialized differently. The performance of the algorithm depends on the complexity of the shadow casters in the scene, because the technique uses a geometric approach. Another problem arises if shadow volumes take in a lot of screen space, or if multiple shadow volumes occupy the same pixels on the screen. In both cases more rasterization-time is needed, as more pixels are influenced or are influenced multiple times. Moreover, shadow volumes have to be clamped if they reach through a wall. Otherwise the shadows will be visible on the other side, where should not affect the scene.

Shadow Mapping

Shadow mapping [52], in contrast to shadow volumes, is an image based approach to compute shadows. It takes advantage of the fact that only points that are visible for the light are lit. This is done by computing a *depth image* for both viewpoints. Then for each pixel in the observer's depth image the depth value is transformed into the view space of the light. The resulting depth value is then compared to the depth value in the depth image of the light (called *shadow map*). If the value is farther away from the light than the corresponding depth value in the shadow map,

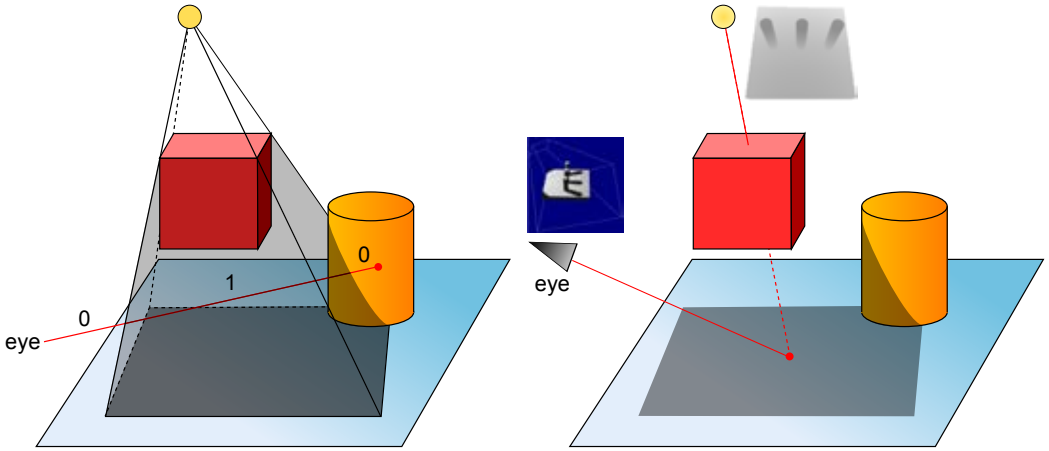

Figure 3.4: The left image shows a ray traveling through a scene with shadow volumes. Every time the ray hits a front-face of a volume the stencil value increases. When a back-face is hit, the value decreases. The right side shows an image where a pixel of the camera is transformed into the shadow map of the light. Because the depth value of the pixel is greater than the one in the shadow map, it is not lit by the light source.

the pixel in the observer's view is not lit by the light source (see Figure 3.4). If a transformed pixel does not lie within the shadow map, it is also not lit.

The transformation is done by using inverse matrices. For instance if we wanted to transform a vector form the *projective space* of the observer, into the *projective space* of the light source, we would apply the inverse projection matrix and the inverse view matrix of the observer to transform the vector into world space. From there the view and projection matrix of the light are used to reach the projective space of the light, where the depth values can be compared (if the transformed vector lies within the shadow map) [45].

Because this technique is image-based, complex objects do not have the same impact on performance as they have for shadow volumes. For each pixel on the screen one test per light source is necessary to determine if a point is in shadow or not. But there can be aliasing artifacts, because the shadow map consists of rectangular pixels that represent a quantized version of the scene. The most common artifacts are *moiré-patterns* and staircase artifacts. As each pixel in world space is orientated in the direction of the view vector, the borders of the pixels have a biased depth value in the depth images of observer and light. So only pixel centers represent the actual depth, while for the rest of the pixels the depth might be wrong (see Figure 3.5). This leads to moiré-patterns (see Figure 3.5). By applying a small bias to the depth values in the shadow map, moiré-patterns can easily be reduced. Staircase artifacts however are caused through perspective aliasing or a too low shadow map resolution and are harder to come by. Simply increasing the resolution also increases computation costs and the amount of memory needed for the shadow map. There has been some research in this area in order to reduce these artifacts [32, 34, 47, 48, 53, 56]. But shadow mapping remains a non-pixel-exact solution, unlike

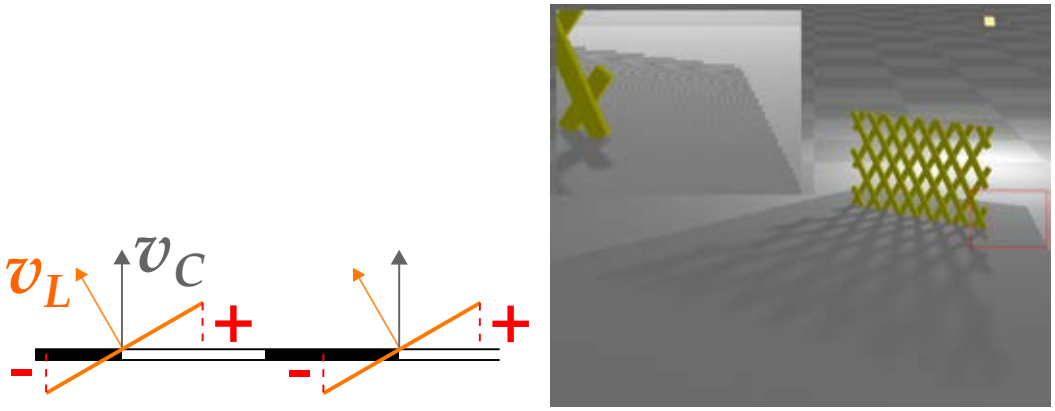

Figure 3.5: Moiré-patterns result from different orientations of camera (black) and light (orange) pixels. The "-" represents areas that are falsely assumed to be in shadow. The picture shows a sample scene with visible moiré-patterns.

shadow volumes. There are, however, features that make shadow mapping suitable for soft shadow algorithms, as will be shown in Section 3.2.

3.2 Real-Time Soft Shadows

The first observation about the difference between soft and hard shadows is that soft shadows have a penumbra. So a solution to produce soft looking shadows is to blur hard shadows with a filter kernel. This can be done by blurring the shadows in the output image, or by using multiple samples of the shadow map to compute the average occlusion of a pixel, called *percentage closer filtering* (PCF) [38]. The difference between these two solutions lies in the estimated size of the fake-penumbras. While filtering in image space produces soft looking shadows of the same blurriness over the whole image for all kind of light sources, the second approach can produce shadows of varying softness for non-directional lights. For these lights the boundaries of a pixel from a frustum. So the size of a pixel grows with its distance from the light. And so does the shadow which fakes growing penumbras.

Though this solutions are not physically based and not a plausible representation for soft shadows, they by design can reduce artifacts and are robust in terms of aliasing. There are other algorithms that reduce aliasing artifacts by *pre-filtering* the shadow map. One is called *variance shadow maps* and was introduced by Donnelly [16] and improved by Lauritzen [30]. Donnelly and Lauritzen use statistical approaches to pre-filter the shadow map by computing the mean and variance of the (linear) depth values in the shadow map. The drawback of these methods are light leaking artifacts that occur at areas where two or more shadows overlap. So in comparison to PCF this solution is not as robust. But both can be used to produce soft looking shadows.

More plausible solutions will be presented in the following subsections. At first we will discuss soft shadow algorithms based on shadow volumes and will talk about the pros and cons of these algorithms. Then we will move on to algorithms that simulate soft shadows by using a

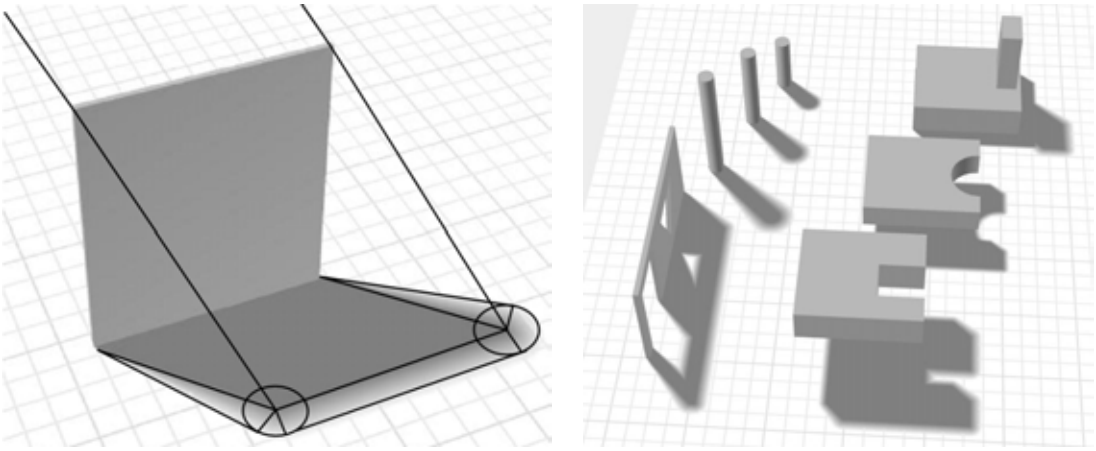

Figure 3.6: To generate planar soft shadows first the umbra is generated by projecting the object onto the plane. Then gradient circles are "drawn" for each umbra vertex and gradient quadrilaterals for edges. (Source [25])

single shadow map. One method has received particular attention in recent years, which is why we will take a closer look at soft shadows by *backprojection* in the last subsection.

Shadow Volume Based

Algorithms that make use of shadow volumes are geometry based approaches. There have also been other geometry based algorithms that produce soft shadows. Especially Haines implementation to generate planar soft shadows [25] is of interest, because it was improved by Akenine-Möller and Assarsson [1] to create soft looking shadows.

The idea behind Haines planar soft shadows is to project the shadow casting objects onto the shadow receiving plane to generate the umbra of the shadow. (The same method was used by Blinn et al. to produce *projected shadows* [6].) The color of the shadow is determined by the vertex colors of the umbra, which can be chosen by the user. Then the algorithm searches for silhouette edges in the shadow caster and projects them onto the ground plane. The projected edges define the silhouette of the umbra on the plane. A circle is generated for each silhouette vertex of the silhouette edges and is connected to its neighboring circles via polygons. The radius of a circle is defined by the ratio between original and projected vertex, and the distance to the light source. The center vertices of the circles lie on the umbra and must have the same color as the umbra. The borders receive a transparent color. The same is true for the polygons. If the colors are now interpolated from the centers to the borders, a soft shadow can be seen (see Figure 3.6).

Because umbra, circles, and polygons are drawn in random order, an approach presented in [35] is used to always choose the darkest shadow where *penumbra shapes* overlap. This results in an overestimated penumbra as the umbra region has the same size as a hard shadow. This means that increasing the size of the area light source only affects the penumbra size while

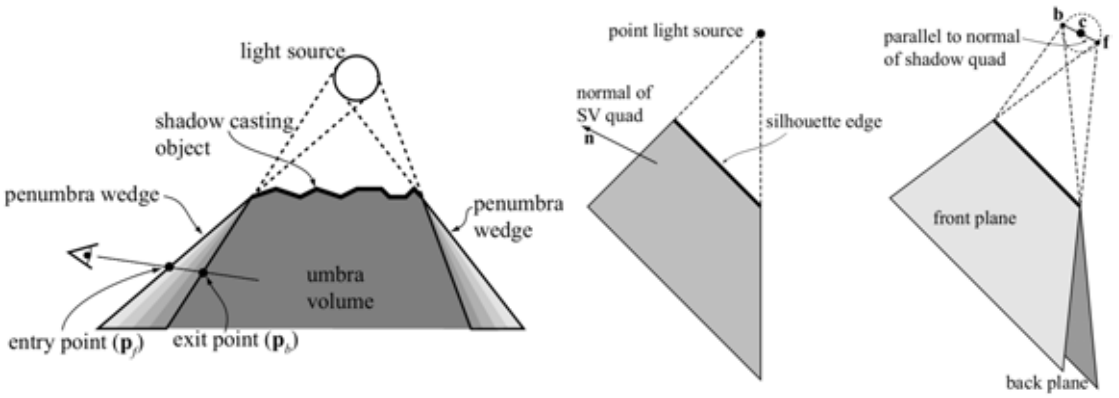

Figure 3.7: The two illustrations on the right show the difference between shadow volume quads of a point light and an area light source. (Source [1])

the umbra stays the same. Another problem is the restricted application area of this method. It can only be used for planar shadow receivers.

To overcome these restrictions, Akenine-Möller and Assarsson [1] combine the ideas of penumbra shapes and shadow volumes. First the silhouette edges from the point of view of the light are determined. Then, in contrast to the shadow volume algorithm, two shadow volume quads are generated for each edge (front and back plane). This is done by using two offset points **b** and **f** that lie on the borders of the light source, as shown in the two illustrations on the right in Figure 3.7. The direction of the offset is determined by the normal vector of the shadow volume quad, which is used for hard shadows. This leads to the following equations for **b** and **f**:

$$\mathbf{b} = \mathbf{c} + r\mathbf{n}$$

$$\mathbf{f} = \mathbf{c} - r\mathbf{n}$$

Where **c** is the center and r the radius of a spherical light source. These two shadow volume quads form a *penumbra wedge*.

In order to compute soft shadows a *light intensity buffer* is used. The buffer is initialized with values of 1 to indicate completely lit pixels. If a ray enters a penumbra wedge at point $\mathbf{p}_f$ and exits at point $\mathbf{p}_b$, the difference of intensity is computed for these two points ($s_{\mathbf{p}_b} - s_{\mathbf{p}_f}$) and added to the light intensity buffer (see Figure 3.7). So for a point $\mathbf{p}$ within the umbra, the intensity to be added to the light intensity buffer is $s_{\mathbf{p}_b} - s_{\mathbf{p}_f} = 0 - 1 = -1$, which means the point is completely in shadow. For a point inside the penumbra we get, for instance, $s_{\mathbf{p}_b} - s_{\mathbf{p}_f} = 0.25 - 1 = -0.75$. The point is 25% lit by the light source. If we enter and then exit the penumbra wedges of a shadow volume, the accumulated change in the light intensity buffer would be $\left(s_{\mathbf{p}_{b1}} - s_{\mathbf{p}_{f1}}\right) + \left(s_{\mathbf{p}_{f2}} - s_{\mathbf{p}_{b2}}\right) = -1 + 1 = 0$. So points behind a shadow volume are lit correctly.

An implementation of penumbra wedges on graphics hardware and optimizations in terms of wedge generation and penumbra estimation were presented in [2] and [3]. These implemen-

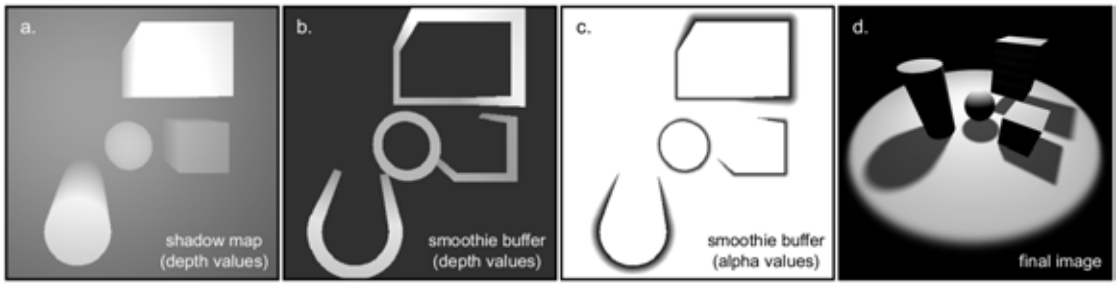

Figure 3.8: Soft shadows with smoothies: a) Create the shadow map from the point of view of the light. b) Create the *smoothie buffer* by rendering the smoothie objects along the silhouettes and store the depth and the ratio between light, smoothie, and receiver distance. d) Render the scene from the observer's point of view and use depth comparison to compute fake soft shadows. (Source [10])

tations produce more plausible looking soft shadows than planar soft shadows. The umbra is estimated smaller than the hard shadow, which is physically more plausible (see Figure 3.1). Though these improvements create better penumbra wedges faster, they still lack the fundamental problem of shadow volume based algorithms: Performance depends on the complexity of the geometry. They are also not physically correct, because silhouette edges are computed by assuming a point light [1] and special cases where penumbra wedges overlap have to be handled differently.

Such cases do not arise in shadow map based algorithms. Moreover there are no problems caused by the complexity of the geometry. But there are other problems to consider, as we will see in the following section.

Single Shadow Map Based

Using shadow maps is a popular way to generate shadows, as everything that can be rendered can be used as an occluder [52]. And as we have already seen, they can be used to fake soft shadows through filtering [38] or pre-filtering the shadow map [16]. We have also discussed that these "simple" approaches are not physically based. In this section we will concentrate on physically more plausible simulations of soft shadows through shadow maps.

For instance, Chan and Durand presented an algorithms which, like planar soft shadows, adds a penumbra to the umbra of the hard shadow [10]. They first render the scene from the point of view of the light to generate a shadow map. Then geometric objects (*smoothies*) are drawn along the objects' silhouettes to create a second shadow map that only consists of smoothy depth values, called the *smoothie buffer*. In addition to the depth values, the ratio between the distance of light source, blocker, and receiver is stored in the smoothie buffer. Finally the scene is rendered from the observer's point of view. Pixels are compared to pixels in the shadow map and the smoothie buffer of the light source. If the pixel is occluded by a pixel in the shadow map, the pixel is inside the shadow. If the pixel is occluded by a pixel in the smoothie buffer,

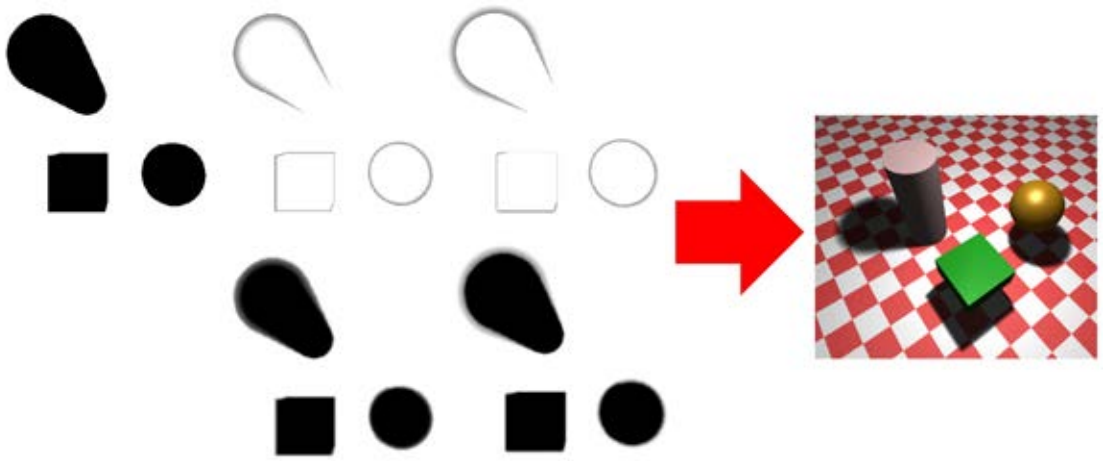

Figure 3.9: Inner and outer penumbra maps are used to subtract and add shadow to the shadow map. (Source [9])

the ratio is used to estimate intensity of the partially shadowed pixel (see Figure 3.8). Similarly to planar soft shadows the umbra is overestimated, because it equals the hard shadow of a point light.

Wyman and Hansen presented a similar algorithm at the same time [54], which also combines the ideas of planar soft shadows with the shadow mapping algorithm. Instead of smoothies they draw cones for each silhouette vertex and sheets connecting adjacent cones. The depth values of these objects are then likewise stored into a separate shadow map which they call *penumbra map*. The difference between these methods lies in the design of smoothie buffer and penumbra map. While the penumbra map only stores depth values, the smoothie buffer additionally pre-computes the ratio between light, blocker, and receiver for intensity calculation. For penumbra maps, this calculation needs to be done in a separate render pass. Because the two methods are so alike, wrong umbra and penumbra estimation, as well as overlapping artifacts of penumbras, are also present in both algorithms.

Kirsch and Doellner tackle the problem of too big umbrae by computing *shadow width maps* [29]. The first step is to generate the shadow map. Then the shadow width map is generated by computing the distance to the nearest lit pixel in the shadow map for each pixel in the shadow width map. To calculate the intensity of a pixel from the observers view, an *attenuation function* is used. This functions returns 1 (fully lit), if the pixel lies outside the umbra of the hard shadow and a value between 0 and 1 for pixels that lie inside the umbra. This effectively reverses the problem of penumbra maps and smoothies, as the soft shadow penumbra can have at most the size of the hard shadow. The algorithm also suffers overlapping artifacts like the previous algorithms.

To achieve more plausibility, the ideas of the penumbra map algorithm can be extended to not only extend umbrae with penumbrae, but also to reduce the umbra. Cai et al. introduce

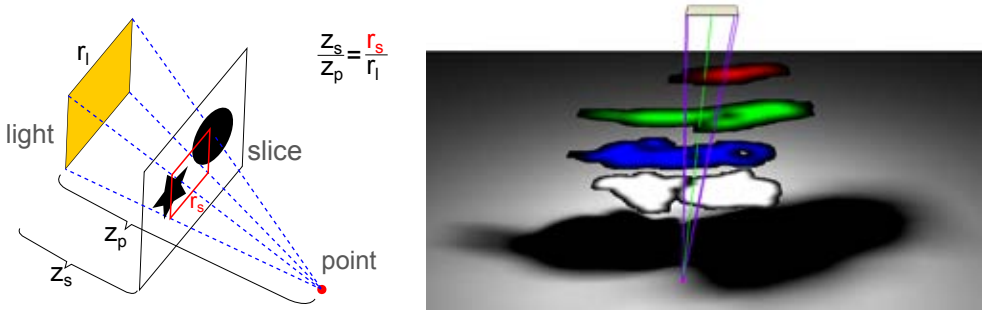

Figure 3.10: A rectangular light source projected onto a plane (left). The plane is perpendicular to the light source an can, for instance, be the near plane of the light view frustum or a texture slice of an occlusion texture. The projection r_s can be computed through *similar triangles*. This is used to determine texture lookup areas in occlusion texture layers (right). (Source [17])

inner and *outer penumbra maps* and *shadow fins* [9]. While outer penumbra maps work the same as in [54], inner penumbra maps are used to "subtract shadow" form the hard shadow. Figure 3.9 visualizes this approach. At first the inner and outer shadow fins are generated from object silhouettes (similar to smoothies and penumbra maps). Then the shadow map, as well as the inner and outer penumbra maps, are rendered form the point of view of the light. In contrast to Wyman and Hansen's approach, Cai's penumbra maps store an additional weight value to indicate a weight of the inner and outer penumbra, which is added or subtracted from the light illuminance. This illuminance is computed when rendering the scene. First normal shadow mapping is performed, where pixels inside the shadow get an illuminance value of 0 and lit pixels a value of 1. If a pixel is inside the shadow, the algorithm adds illuminance from the inner penumbra map. If it lies outside, illuminance from the outer penumbra map is subtracted. Though this increases plausibility, overlapping artifacts are still a problem. To reduce this effect, Cai et al. purpose a multi-layered version of their algorithm. This is done by dividing the view frustum of the light into multiple layers. Then the illuminance is computed for each layer. To increase the rendering speed, subsequent layers use the information of previous layers to calculate the illuminance. This reuse reduces light leaking artifacts, if the factor of illuminance is chosen carefully between the current and the previous layer. For each pixel in the final image the right layer must be chosen depending on where it resides in the light frustum.

This is similar to Eisemann's and Décoret's soft shadow solution [17]. They too divide the light view frustum into layers called *slices*. But instead of storing information about depth and weights for umbra and penumbra, they only store information about weather a pixel contains geometry or not. For each slice the *occlusion texture* indicates where subsequent pixels in farther away slices are occluded. Because this would only project a binary texture onto the geometry of the next slice, each texture slice needs to be filtered to achieve soft looking shadows. In order to achieve plausible results the filter size must be chosen with respect to the distance between light source and slices. Slices closer to the light source need bigger filters, because the geometry lies closer to the area light. When the final image is rendered, all slices between point and

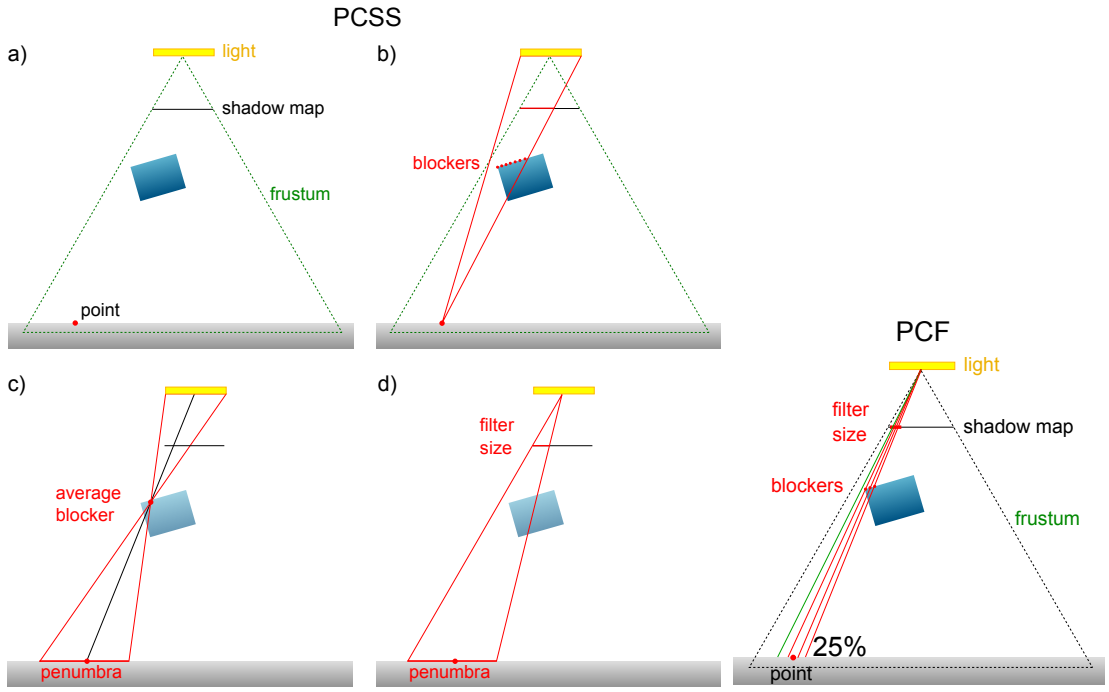

Figure 3.11: The left image shows PCSS algorithm: a) First render a shadow map. b) Then search for blockers in the shadow map for a point in the scene. Use *similar triangles* to compute search area by projecting the light source onto the near plane of the light frustum. c) Average the depth values of the blockers and estimate the size of the penumbra through *similar triangles*. d) Use *similar triangles* math again to compute the filter size for PCF. For PCF (right image) multiple samples of the shadow map are selected depending on the filter size. The amount of light reaching the point is defined by the number of non-blocking samples.

light source have to be accumulated into a single intensity value for each point. This involves a texture lookup for each slice, depending on the projection of light on the slice (see Figure 3.10). Eisemann's and Décoret's use a specific accumulation function to get smooth transitions between different layers [17]. The disadvantages are comparable to the shadow fins algorithm. There is missing self-shadowing inside a slice, because only previous slices influence a point. To omit light leaking artifacts efficiently, occlusion textures are projected onto their successors.

A drawback in terms of performance for the last two techniques is that multiple layers/slices have to be created to reduce artifacts. It is also necessary to implement sophisticated accumulation mechanisms. This makes these methods hard to implement when comparing them to techniques like PCF. But PCF does not estimate penumbra sizes and is not a physically plausible solution. So the question is how to vary the filter size depending on the size of the penumbra at a specific point in a scene. One solution was presented in [19], which is called *percentage closer soft shadows* (PCSS). PCSS has several advantages compared to other solutions: It is independent from the geometric complexity of the scene, as it is based on shadow mapping, and

it is not necessary to pre-process or post-process the shadow map or the rendered image or to generate additional geometry. Moreover, it is easy to implement, because it only extends the PCF algorithm by *similar triangle calculations*, as can be seen in Figures 3.10 and 3.11. For the algorithm a shadow map has to be rendered from the viewpoint of the light (Figure 3.11a). For each point rendered from the observer's point of view in the final image, the light source is projected (using similar triangles) onto the near plane of its view frustum with respect to the point. The projected area determines those pixels in the shadow map that have to be checked for *blockers*. Blockers are pixels that occlude the light source (Figure 3.11b). Because this area can vary in size for each point in the scene and can include all pixels in the shadow map, stochastic sampling [11] is used to reduce the number of tests. Once the blockers are found, the average blocker is computed (Figure 3.11c). This allows an efficient estimation of the penumbra. Note that this is not a physically correct approach, but satisfies the physical properties of soft shadows. Where objects contact each other shadows are sharp and grow softer with increasing distance between occluder and receiver. From the estimated penumbra size the filter size for PCF can be calculated. Again similar triangles are used (Figur 3.11d).

This approximation of the penumbra produces believable results [19]. However, it suffers from wrong penumbra estimation through the averaging of the blocker distance. If several objects are significantly separated in space, but lie inside the search area for the blocker search, each of the objects influences the outcome of the average blocker. More realistic results can be achieved, if only those blocking pixels where used that affect the point of interest. This approach is used in the *backprojection* algorithm.

3.3 Soft Shadows Using Backprojection

There has been some research on soft shadows using *backprojection* and there are several different implementations of this method. Atty et al. purposed that for each occluding pixel in the shadow map a soft shadow should be created and accumulated into a single *soft shadow map* [4]. This involves the following steps:

1) Divided objects into occluders and receivers and generated a shadow map for both groups.

2) Compute the representation of the pixel quads in the scene form the occluder shadow map (*micro-patches*). This creates a discretized version of the occluders.

3) Calculate the penumbra extent for each *micro-patch* (Figure 3.13).

4) For each point on the receiver objects inside the penumbra extend, calculate the percentage of occlusion by projecting the micro-patches onto the light (Figure 3.13). The sum of all points generates the *soft shadow map* 3.12).

To compute the soft shadow map, the fact that light source and micro-patches are perpendicular is used. This way micro-patches can be efficiently projected back onto the light source by simple math [4] (see Figure 3.13). Then the projected area is clamped by the size of the light. Now the percentage of the *backprojected* micro-patch on the light source can be subtracted from

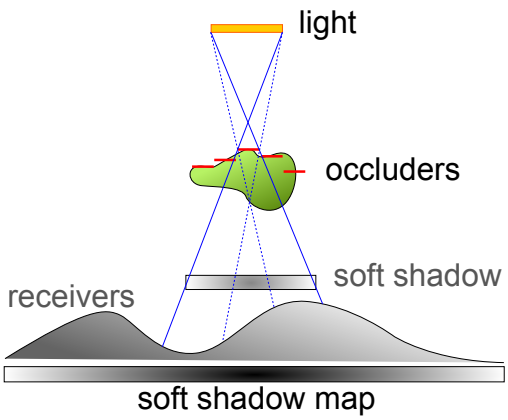

Figure 3.12: Create micro-patches from the occluder shadow map. Calculate the penumbra extend for each patch and sum all soft shadows created by the patches into a single *soft shadow map*.

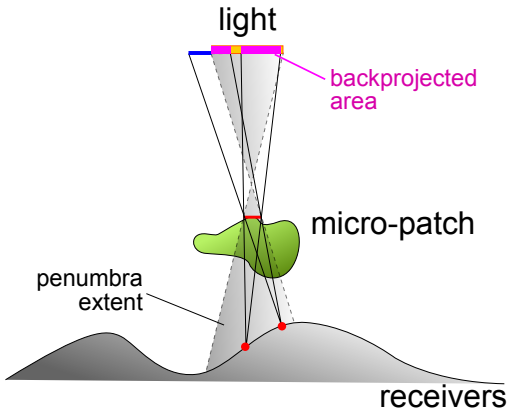

Figure 3.13: Compute the amount of light that is not blocked by the micro-patch for each receiver point inside a penumbra extend. All points together form a soft shadow for the micro-patch. All soft shadows of all micro-patches form the soft shadow map.

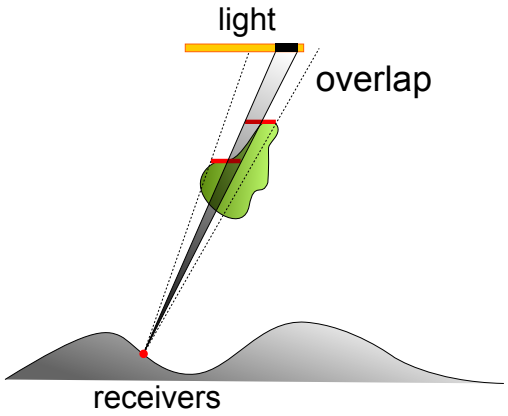

Figure 3.14: Overshadowing artifacts are caused by overlapping backprojected micro-patches.

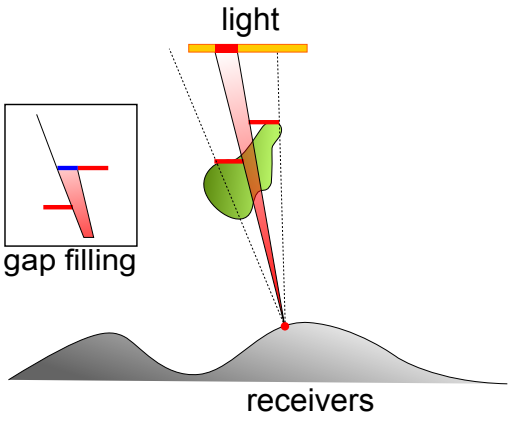

Figure 3.15: Light leaks are caused by gaps in the discretized occluders. To fill gaps, *gap filling* can be used.

the intensity that would illuminate the receiver point. Taking a look at Figure 3.14, we see that backprojected areas on the light source can overlap. So the same area of the light source might get subtracted several times, which makes shadows darker than they should be. There can also be light leaks, if there are gaps between separate micro-patches, as shown in Figure 3.15. These errors are caused by discretization of occluders into patches. Especially light leaks lead to prominent artifacts (see Figure 3.16).

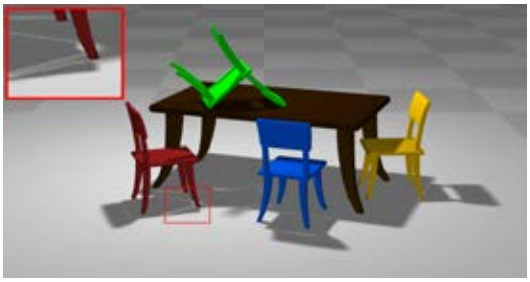

Figure 3.16: Light leaking artifacts caused by gaps between micro-patches have a negative influence on the plausibility of a soft shadow. Such errors can be reduced by applying *gap filling*.

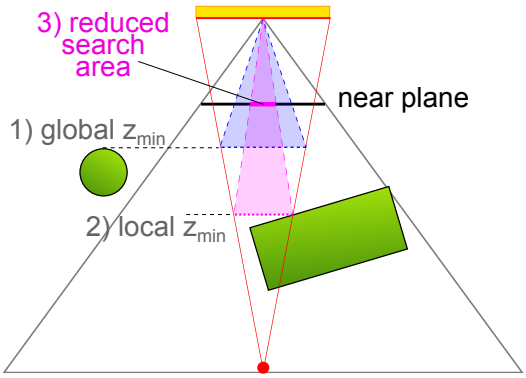

Figure 3.17: Reduce the search area by first using the global depth minimum $global z_{min}$ (highest level in the HSM) and then the local $local z_{min}$ (HSM layer defined by global depth minimum).

Guennebaud et al. reduce light leaks by extending the micro-patches to close gaps [23]. This approach is called *gap filling*. The micro-patch size is adjusted by backprojecting itself and its neighboring patches (e.g. left and bottom) onto the light source and then choosing the minimal value of neighboring edges (see Figure 3.15). Besides removing the restriction of differentiating between occluders and receivers, no soft shadow map has to be computed, which reduces memory usage. Instead the intensity reaching a point is estimated from the observer's viewpoint. Their approach is similar to the blocker search step of the PCSS algorithm, but the search area is additionally reduced by generating a *hierarchical shadow map* (HSM). The HSM stores the minimum value for each 2x2 pixel block in the shadow map for each subsequent level. The reduction of the search area is achieved by utilizing the fact that blockers, which affect the point, must lie inside the pyramid between point and light source. So if we know that there is a minimal depth value in the pyramid, all other potentially blocking micro-patches must lie behind this depth inside the pyramid. This in turn means that if we slice the pyramid at this depth, the depth values of potential blockers must lie inside the projection of the slice onto the near plane of the light frustum. By using the top level of the HSM we get the minimal depth value in the shadow map. This lets us compute a first reduced search area, as shown in [23]. With this area we can select the local minimal depth value by choosing the appropriate mipmap level in the HSM through the size of the reduced search area. This reduces the area even more as shown in Figure 3.17.

Artifacts remain in terms of overshadowing, because gap filling only handles light leaking artifacts. Schwarz and Stamminger purpose the use of *bitmask soft shadows* to tackle this problem [42]. They also purpose different representations of micro-occluders [43] and blocker search methods [44] to further increase the soft shadow quality.

3.4 Comparison

We have discussed several real-time soft shadow algorithms which can be divided into shadow volume and shadow mapping based methods. The main performance bottleneck of shadow volume techniques is the need to analyze the occlude geometry before the shadows can be generated. Overlapping penumbras have to be handled by adjusting the shadow volume geometry, which costs additional computation time.

Shadow mapping does not need additional geometry processing to find silhouettes. Though some shadow map algorithms also generate additional geometry from silhouette edges (e.g. smoothies or shadow fins). There is no need to distinguish between occluders and receivers (except for the original Backprojection algorithm). On the other hand, the scene needs to be rendered in an additional pass, form the lights point of view. So there is additional cost to compute object visibility and to render objects again for each light.

In terms of robustness and artifacts, there are no algorithms not suffering from penumbra overlapping artifacts. Each algorithm has its flaws. Some algorithms divide the light view frustum into layers to improves plausibility of the results and reduce artifacts. However, these solutions increase the cost for rendering and make the implementation more challenging. The scene also needs to be divided properly so artifacts are removed, making these methods unattractive for arbitrary scenes. When compared to each other, shadow mapping based algorithms that only need a single shadow map and no additional data structures, are the easiest to implement. They only add additional calculations and shadow map lookups to the basic shadow mapping algorithm.

In the next chapter we will discuss which algorithms were chosen for the study based on the overview given in this chapter.

CHAPTER 4

Experiment

Because we want to find out which shadow algorithms are more realistic than others and which factors influence the decisions of an observer, this chapter will describe the algorithms used for the study. We will also take a look at important factors that influenced the selection of stimuli. Then the block design and structure of the study will be described. In the end the setup for the automated generation process of statistical data is explained.

4.1 Used Soft Shadow Algorithms

Because of the high number of conceptually different soft shadow algorithms and the high implementation effort involved, we decided to restrict our survey to four representative algorithms, which span the whole range from simple but heuristic, to costly but fully physical. Most other soft shadow methods can then be placed somewhere within this range. Since shadow volumes have a number of practical problems, like consuming a lot of GPU fill rate and being seldom used nowadays, we selected our four algorithms solely among those for shadow mapping. The algorithms were chosen because they represent *distinct classes* of shadow-map based soft shadow algorithms. For all algorithms we decided to avoid artifacts like *overshadowing* or *staircase artifacts*, because the presence of artifacts resulted in a clear preference for artifact-free images in our pilot study (see Chapter 4.3). These four algorithms are:

Percentage-Closer Filtering (PCF) This method has been originally proposed as an anti-aliasing method for shadow borders, by filtering the binary shadow map test results [38]. However, it can also be used to simulate the penumbra regions of soft shadows (i.e., the softness depends on the kernel size). As the kernel size is fixed for a given shadow map resolution, the method does not capture contact hardening nor allows for penumbra size estimation (as can be seen in Figure 4.1). Hence the plausibility of this algorithm is the lowest of all four, but it performs faster than the others. Because graphics hardware already provides a filtering mode for depth textures that include a simple version of this method, it is widely used.

Percentage-Closer Soft Shadows (PCSS) This is an example for a single-sample soft shadow algorithm. PCSS is based on PCF, but here the size of the penumbra is computed dynamically, based on the relative distance of a receiver from the light source and the blocker geometry. Hence it is more physically founded, but requires more computations and texture look-ups (for the blocker search) than the PCF algorithm.

Backprojection Backprojection [4] estimates the amount of light reaching a point in the scene by identifying small blocker patches from the shadow map and backprojecting them on the light source. This method is more physically based than PCSS (it can capture contact hardening), but at the same time costlier. Unfortunately, this method is prone to light leaks and overshadowing. Our implementation uses a gap-filling approach [23] to reduce the light leaks. Nevertheless, it turned out to be quite difficult to avoid artifacts by tuning the considerable amount of parameters of the algorithm. Furthermore, it requires high implementation and optimization effort, e.g., for accelerating the blocker search. Note that several methods have been proposed to make Backprojection more robust [42, 43].

Ground Truth We use a brute-force method as the reference solution, which simply samples the area-light source multiple times and then blends together the resulting shadow maps. As this method converges to the physical ground truth for a sufficient number of point lights, we call it the ground truth solution within the scope of this study. In our experiments the solution converged after 1024 samples (i.e., there were no more changes of pixel color).

4.2 Important Factors

There are factors that potentially influence the perception of soft shadows: observer and object movement (animation), texture color and complexity, or the current context (e.g., walking from one point to another versus fighting in a battle in a game). Especially animation can play an important role by hiding artifacts if the shadow changes faster than it can be evaluated. Jarabo et al. [26] were able to show that random movement and increased complexity of animated objects make errors less noticeable for users. In this study we focus on the following factors: complexity of shadow casters and receivers, their spatial relationship, and overlapping and varying penumbras. We decided to limit this study to these factors, because the experiment was already exhaustive without them (60 minutes duration). For the remaining factors, we choose "worst-case" scenarios: First, observers and objects will not move, so that the shadow can be easily inspected by the user. Second, shadow receivers will have a white texture to ensure maximum contrast. In order to allow participants to concentrate on the evaluation of shadow quality, they are placed in an observer situation with no distractions (like enemies).

Scene Characteristics

In order for the experiment to be meaningful, the chosen scenes have to include a variety of object and shadow complexities. After extensive pre-studies, 13 categories (12 systematic and 1 real-world) were defined. Each category is represented by 2 different scenes (see Figure A.6 for

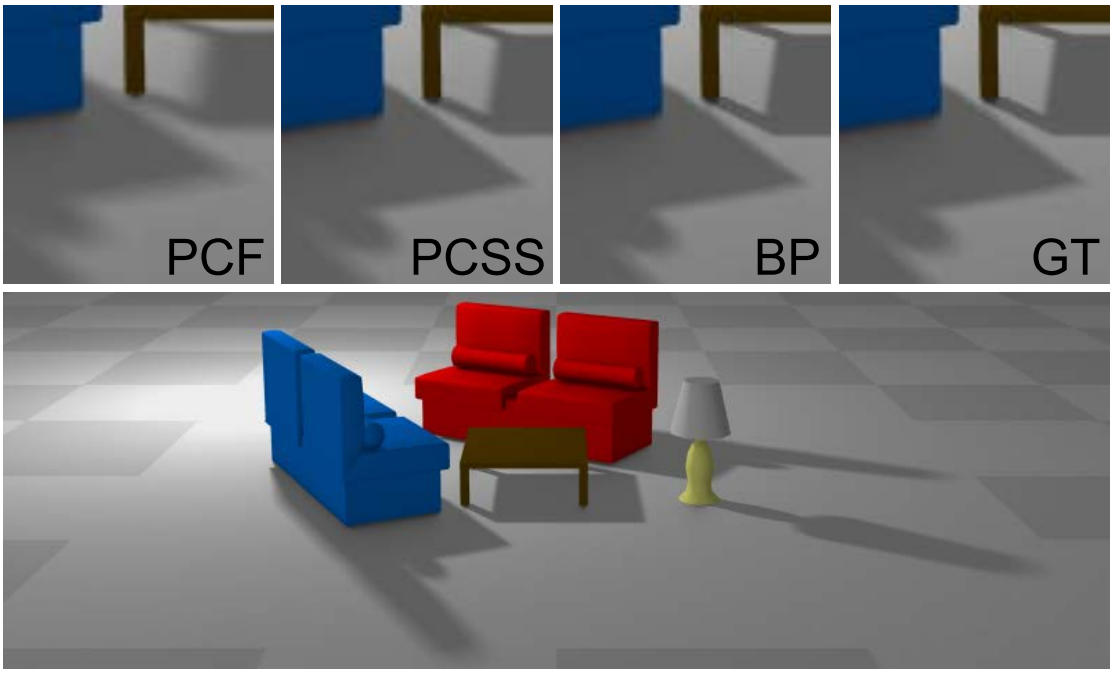

Figure 4.1: Closeup of soft shadow properties. PCF cannot capture penumbra variation and contact hardening of soft shadows. PCSS handles contact shadows from the table leg but is a crude approximation. Backprojection (BP) is almost indistinguishable from the reference solution (GT).

images of all categories and scenes). Categories are defined following two criteria: the spatial relation of light and objects in the scene, and the complexity of the scene (see Figure 4.2).

Spatial Relation of Light and Objects

The spatial relation of light and objects can cause two effects: First, shadows can overlap because of multiple objects or self-shadowing, and second, the penumbra varies or does not vary in its size (see Figure 4.1 for an example of a varying penumbra), resulting in 4 subcategories.

Scene Complexity

There are several factors that influence scene complexity, like the number of objects in the scene and their respective complexity, the distribution of objects, and the regularity of objects (e.g. a fence consists of planks). Moreover, complexity is a subjective parameter and can be perceived differently by different people. So we decided to create a list of observations that influence the appearance of a scene and the shadows in it. As shadows are a direct result of objects casting shadows, we first make observations about objects in a scene:

1. A scene can contain one or more objects.

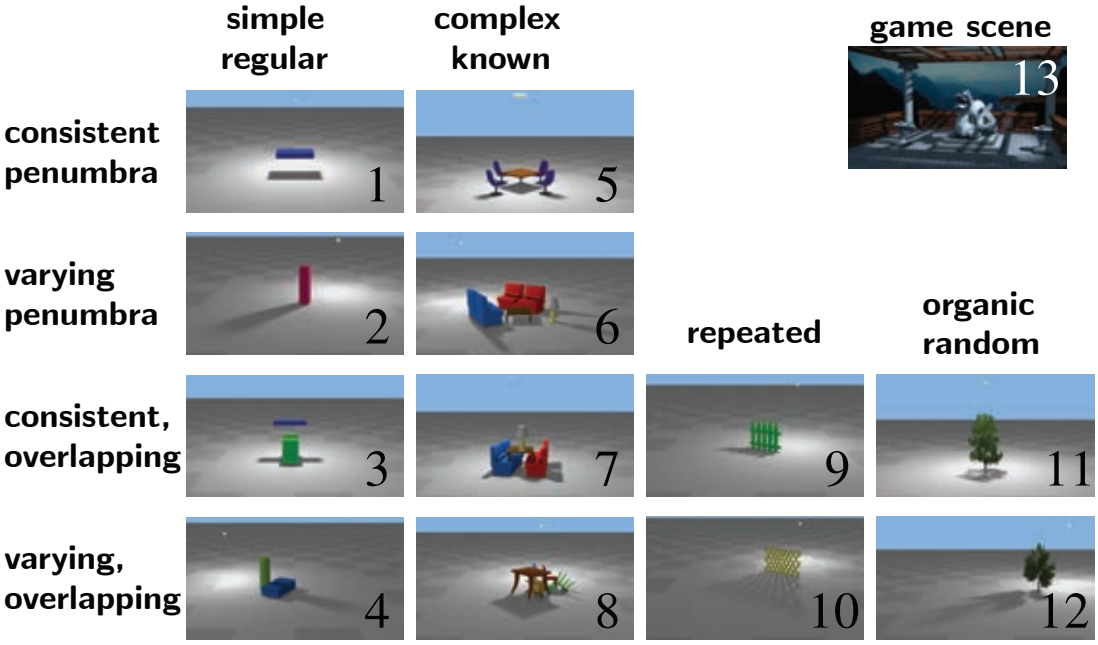

Figure 4.2: This category matrix defines 13 scene categories. An empty field marks a category that is covered by another.

2. Simple objects can be arranged to form complex objects (e.g., a table or a chair).

3. Simple objects can be arranged in irregular patterns to form complex objects (e.g., a tree).

4. Simple objects can be arranged in regular patterns to form complex objects (e.g., a fence).

Based on these observations we can now define four object subcategories:

1. Simple, regular objects like boxes and cylinders. These objects exhibit strong symmetries, which might ease comprehension of the shadows for the participants. In applications, such objects are for example used to represent roofless houses, skyscrapers, or columns in urban environments (Categories $1 - 4$).

2. Assemblies of objects of the first category. In particular, we used simple furniture items, as these are easily recognizable by participants and therefore do not draw attention away from the task of evaluating shadows (Categories $5 - 8$).

3. Regular assemblies of objects along a plane or line. For example, fences can be seen as a regular arrangement of boxes on a line and the crossbars of the fence as regular alignment of boxes on another line orthogonal to the first line. These objects can cause artifacts in shadow algorithms like PCF, PCSS and Backprojection if the resolution of the shadow

map is too low. Note that an additional condition for this subcategory is that regularly arranged parts of an object have to be connected in order to ensure that their shadows are perceived in combination and not as several individual shadows (Categories $9 - 10$).

4. Organic objects like plants. They are an irregular and random arrangement of smaller objects (e.g., leaves) in 3D space. Because of the random angles and sizes of leaves and twigs, they cause artifacts in shadow algorithms (Categories $11 - 12$).

We can also make observations about shadows in a scene:

1. Shadows of different objects can overlap.
2. Shadows of the same object can overlap.
3. Depending on the spatial relation between light and object the size of the penumbra varies.
4. Shadows from different objects that do not overlap can be judged separately, as they do not influence each other.

This allows us to define four shadow categories that are a combination of two factors: The appearance of the penumbra (how much it varies in a shadow) and occurring overlaps (if there are any or not). So the shadow complexity of a scene with one light source is defined by object and shadow complexity.

Combining the Criteria

Simply combining the two criteria (subcategories) would give 16 different categories. However, there are some special cases: Fences, grids and plants always have overlapping penumbras in their shadows (see definition of object subcategories in Chapter 4.2). Therefore there cannot be a spatial subcategory with non-overlapping shadows in these spatial categories. Additionally, organic random objects typically have overlap in their penumbras, which further reduces the possible combinations to 12. Finally, to also investigate a realistic setting, we added another category (a game scene), with varying, overlapping shadows, resulting in a total of 13 categories.

Other Scene Characteristics

There are constants parts in the scenes. The first is the ground, which receives the shadow. Because the experiment is a worst-case study, a white plane was used for the ground. There is no unevenness in a plane, which might have made it harder for the participants to investigate the shadows. At first a white material was used for the plane. This, however, made it difficult to comprehend the perspective in the scene, and the orientation of the plane. To improve this, a checkerboard texture was used. But as the study does not take textures into account, a white area in the center of the plane was reserved to receive the shadow.

For the light, a white color was chosen and for the ambient term 0.4. Not using an ambient term would have made the shadows black and the grid invisible. A low ambient of 0.2 would have made the scenes look rather gloomy and a higher ambient terms would have reduced the

contrast of the shadows. For the objects in the scenes, diffuse shading on a per pixel basis was used in addition to the ambient term. For the background a light blue color was chosen.

Because textures were not applied to shadow objects (except for the ground), objects looked flat, as the light source was always placed to cast the shadow into the direction of the observer. So the participants only saw the ambient backside of the objects. To increase the plasticity of the objects, ambient occlusion was precomputed for each scene with 3dsmax. The settings were chosen so that ambient occlusion had only a subtle effect on the scene and no artifacts were visible. To provide a clear distinction between shadow-casting objects and the ground plane, objects received primary and secondary colors.

User Experience with Soft Shadows

A common method to evaluate the quality of rendering methods is to perform a user study where users have to do pairwise comparisons between two images rendered with two different methods. The most conservative method is to provide a ground truth reference. This comes with the advantage that the evaluation task, to estimate for both images the perceived difference to the ground truth, is clearly defined. However, from a practical point of view it is a rather artificial case that a ground truth reference is available. Moreover, similarity to a ground truth does not necessarily scale with the aesthetic appearance.

Without a ground truth reference at hand, a user has to predict how a soft shadow looks like. This is a complex task and requires experience or even knowledge about the physics of light and shadows. As observed in pilot testing, asking users about the degree of realism of a soft shadow approximation may result in an indifferent judgment or even frustration about the difficulty of this task. For most users it is more intuitive to perform an aesthetic judgment by asking which of the two shadows "looks better". However, this comes with the disadvantage that we only have an intuitive guess of a user being clueless about how soft shadows should look like.

Overall, both methods described above may produce different results, which are also fairly limited in their explanatory power. Thus we designed a paradigm which tries to evaluate the perception of soft shadow approximation at different levels of user experience. We achieve this be reusing the knowledge participants gain during the study, and put it to use by evaluating multiple levels of experience. In contrast to other approaches, our novel concept allows us to capture the cases of *inexperienced*, *experienced*, and *expert* users which are defined as follows:

The inexperienced user: The inexperienced user is assumed to be a participant who arrives at the experiment without any skills or previous knowledge about how to judge the realism of soft shadow algorithms. He or she is not provided with any ground truth reference and can only judge on intuition. The instruction is to compare two images in each trial and judge which of both looks better. This user represents people without experienced in computer graphics and rendering. This includes casual players of computer games.

The expert user: The expert user has optimal knowledge how a physically correct soft shadow should look like. This situation is simulated by providing a participant with a ground truth reference for direct comparison. This user represents the worst case of a user who

has optimum sensitivity to validations of physical laws in light rendering. It may include computer graphic experts working in the field of rendering.

The experienced user: The experienced user is a user who has learned how soft shadows should look like. Unlike the *expert user*, the experienced user has no access to the ground truth and hence has to judge based on previous knowledge. Learning has happened during the preceding block of the experiment, where the participant had to compare soft shadow approximations with a ground truth reference. This user represents a person with knowledge about realistic painting, computer graphics or the physics of light. It may include visual artists, heavy computer game players with a great interest in rendering techniques, and experts in computer graphics or the physics of light.

These levels are represented by three consecutive blocks in the experiment which all participants run through. In the first block, we collect data for the *inexperienced user*, in the second for the *expert user* and in the third for the *experienced user*. Learning how soft shadows actually look like takes place in the second block in which an "inexperienced" participant becomes an "experienced" participant.

4.3 Design of the Experiment

The experiment was carried out in a room with blacked out windows to ensure the same pure artificial light conditions for each participant. The procedure was divided into three blocks, separated by two short recreational breaks. Each block comprised in total 91 trials. In each such trial a participant had to compare a pair of images (see Figure 4.3). To assure a sufficient level of motivation, each participant was paid a reimbursement of 10 Euro.

Comparison Task

The participants were instructed that they would see two images, which had to be compared by judging the soft shadows cast on the floor of the scene. They were also told that the light source, producing the shadow, is an area light source visualized as a yellow square on its respective position. Each test subject had to participate in all 3 blocks in the specified order (see Figure 4.4).

Block I The participant was asked to choose the image where he/she thinks the soft shadow looks better. The reason for querying the aesthetic properties instead of the physical plausibility was that without a reference solution, judging the physical correctness might be a too difficult question (especially for people without prior training).

Block II A reference image was shown on top of the image pair to be compared. The participants were informed that this is the physically correct simulation. They were then asked to judge which of the two images (below the reference) models the physical properties of the soft shadows more plausibly. Our expectation was that this block stimulates a learning process by making participants sensitive to the properties of physically correct simulations. It also makes them experts, because the reference image shows them how

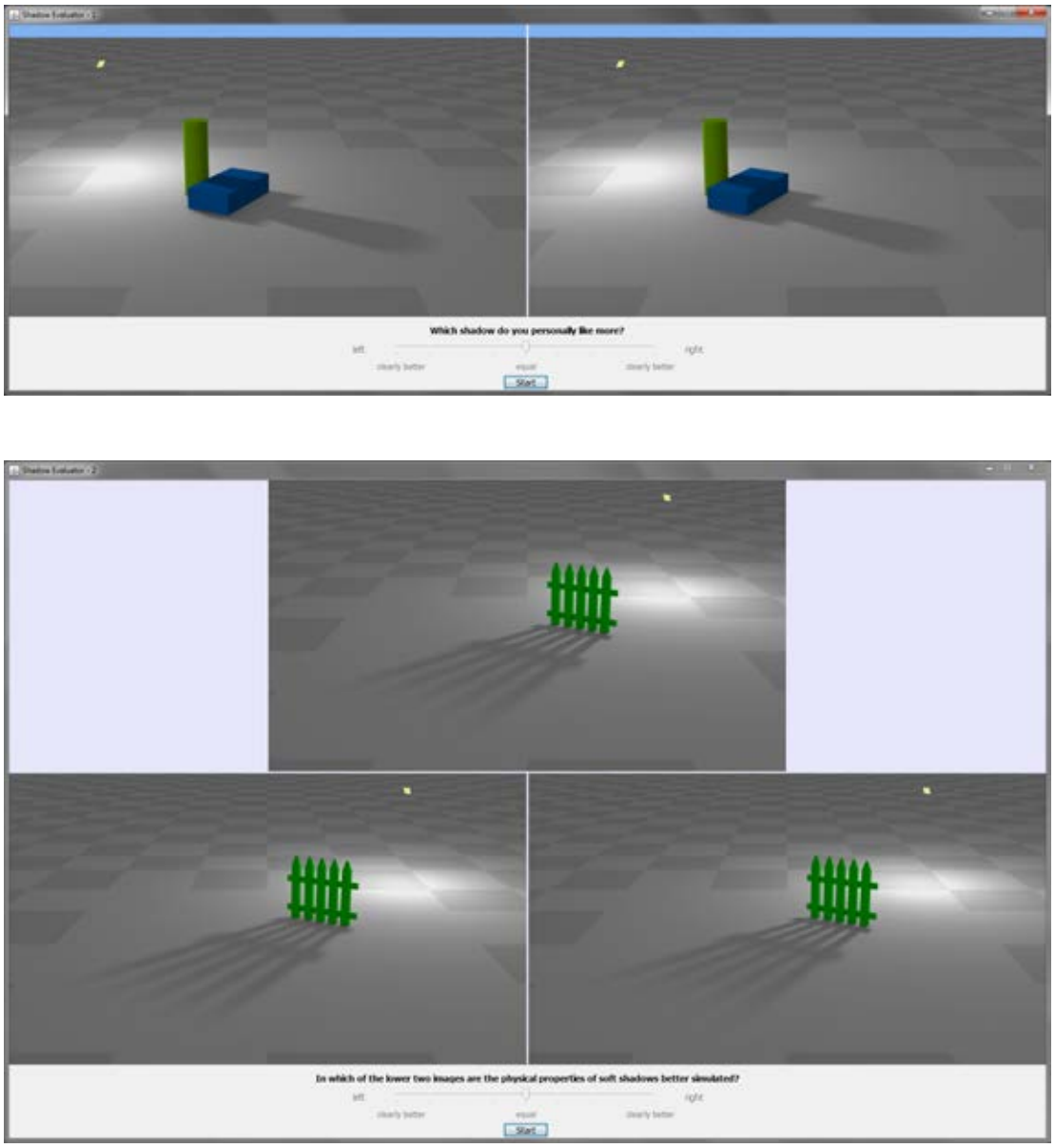

Figure 4.3: The setup which lets the participants cast votes. While Blocks I and III compare two images, Block II additionally shows the ground truth solution as a reference on top.

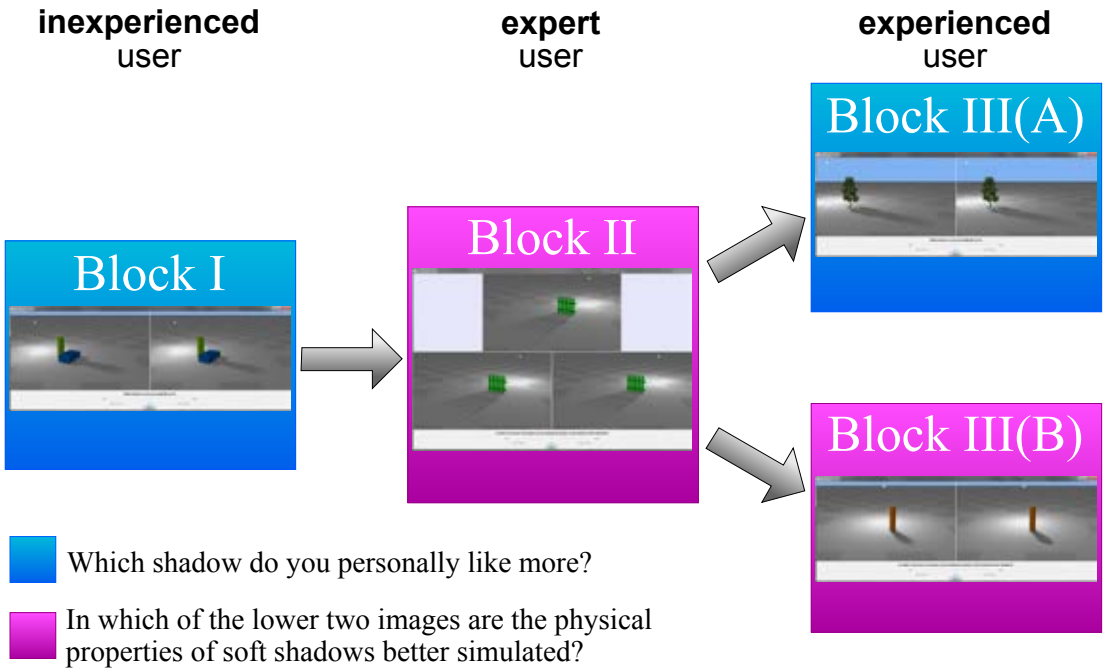

Figure 4.4: The study is divided into three blocks. In Block I inexperienced users are asked which shadow they like more. In Block II an image of the correct soft shadow is shown to simulate expert users. In this Block users are asked about physical plausibility of soft shadows. Block III is divided into group A and B. Group A is asked which shadow they like more while group B is asked about the physical plausibility. This block represents experienced users.

the correct soft shadow should look like. An image of the setup in Block II is shown in Figure 4.3.

Block III This block was without a reference image as in Block I. Participants were divided into two groups with female and male subjects being evenly distributed over both. *Group A* received the same question as in Block I (which shadow looks better), and *Group B* the same question as in Block II (which shadow is physically more plausible). Note that asking about physical plausibility was justified at this point, as the participants already went through a learning phase.

A response was given with a slider, where participants could score the degree of preference for the left or the right image of a pair. The slider had a continuous scale. On the left and right ends, labels were placed defining a clear preference for the corresponding image. At the center of the slider, which was located between both images, there was a mark defining no preference. To provide a distinct option for an indifferent response, the slider had a snapping mechanism at this point. Each participant was told about this behavior and that it is definitely possible that two images are identical or of similar quality.

The decision against a 2-options forced choice paradigm was motivated by our concern for cases, where the actual perceivable difference between the results of two algorithms was low. For these cases, the paradigm would have had the side effect that participants might base their decisions on other factors than those investigated. If such a factor (e.g., a general preference for sharp edges) causes systematic effects, this may have resulted in contradicting responses and lowered the consistence of the results (see Section 5.2). Moreover, from responses obtained by pairwise comparisons alone we could not evaluate, if a small difference in the result was due to a low precision of the response behavior, or due to an actually low contrast in the quality of the algorithms being compared. These small differences are evident as low agreement between the responses of different participants (see Section 5.2). Observing also the amount of no preference responses gave us a hint whether the first or latter reason applied.

Note that a continuous response scale was chosen instead of an ordinal one to reduce the issue of predominantly neutral or nearly neutral responses. The continuous scale is supposed to give participants the illusion to specify the magnitude of their preference, whereas the data was going to be analyzed on an ordinal scale ("left preferred","no preference","right preferred").

For every comparison, a participant had as much time as needed. The next trial was launched after the response to the previous and was preceded by a black fixation cross on a white background which was displayed for 2000 ms.

Image Pairs

The 91 image pairs were created by combining images rendered with one of the four soft shadow methods for each of the 13 scene categories. Building $\binom{4}{2} = 6$ possible pairings of the 4 soft-shadow algorithms for each of the 13 categories. Which in turn yielded in 78 pairwise comparisons per participant. In the other 13 trials we showed one pair of identical images for each category, rendered with one randomly chosen algorithm. This method was used to make participants aware that there was the possibility for images to be equal. This data was collected to obtain a baseline for the participants' response and their agreement for identical image pair conditions.

To have a minimum variety in each category, we created two scenes for each of the 13 categories, which were obtained by different arrangements or by using different models. All images used in the experiment were generated offline by rendering each scene with the four soft-shadow algorithms. All four soft-shadow methods utilized a shadow map of 1024x1024 pixels with 32 bit floating-point precision. The configurations of the light sources (color, intensity, and size of the shining area) as well as the ambient term were kept equal for all rendering methods. Other rendering parameters were adjusted such that the resulting images resembled the ground truth as much as possible. In scenes with strongly varying penumbras, we needed to set the parameters for PCF such that the penumbras lie between the smallest and the largest penumbra of the ground truth. Moreover, we decided to focus on the cognitive ability of users to judge a soft shadow approximation rather than observing their ability to detect artifacts (which is a different perceptual task). Hence, we attempted to avoid artifacts by tweaking the parameters accordingly. This was an important necessity, as we could observe in our pilot study that artifacts greatly overrule all other factors. The presence of one artifact, for instance, may result in a clear preference for the artifact-free image, unless the physical correctness of the shadow in this

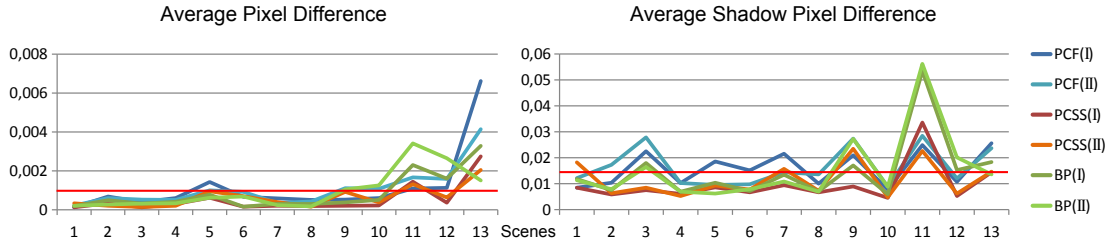

Figure 4.5: The average pixel difference of each image and each scene. The values lie in the interval $[0, 1]$, where 0 means there is no difference between pixels. The higher the value, the greater the difference. If one image were black and the other white, we would get 1. The red lines indicate the mean average pixel difference of all categories.

image is much worse. Finding an artifact-free configuration was particularly difficult for the Backprojection algorithm. High priority was also taken in ensuring that PCF penumbras did not contain too noticeable light leaking artifacts (see Figure 4.1).

Avoiding Ordering Effects

To counterbalance time-related effects, all participants saw an individual randomized order of image pairs throughout all blocks. The images of each comparison pair were randomly assigned to the left or right position. In Block I, participants saw one of the two scenes we created for each category and in Block II the other. Which of both scenes was selected for which block was also randomly determined for each category and participant. In Block III, image-pairs were picked randomly for both scenes of a category such that a participant saw 50% of the scenes in the first block and 50% in the second, respectively.

Because of this sophisticated design we needed to make sure that the two images used for each category produced usable and valid data. Two images are valid, if they have similar shadow properties. We therefore analyzed the pixel difference over the whole image and over those parts that were affected by soft shadows as shown in Figure 4.5.

The average pixel difference over the whole images and the shadowed areas are low, with a mean of 0.09% and 1.4% respectively. We can also see in Figure 4.5 that the two images generated for each category lie close together in terms of average pixel difference. Thus there is approximately the same amount of shadow difference in each of the two images for each category and images of each category have similar shadow properties. This makes them valid stimuli.

4.4 Experiment Setup

The setup of the experiment can be split into three main steps: The first step is to generate the samples to be evaluated by the participants and the second, the execution of the experiment itself. The last part of the setup is to generate scripts of the gathered data for statistical evaluation.

For the first step it would have been possible to compute the sample images in real-time during the evaluation. The problem with this solution was that the evaluation-program would run slower when showing a ground truth sample. So participants could have guessed that the ground truth algorithm was used if the slider did not respond immediately. The pre-generation of the sample images made it possible to use different hardware for the generation and evaluation of the samples. And they could be used multiple times without being changed by switching graphics hardware or updating drivers. Only the monitor could influence the appearance of the images.

Generation of Samples

In order to allow an automated generation of images a program was written. Generated images were stored in a folder on the hard drive. This way it was possible to quickly recreate the sample images if changes occurred. The program also provided a possibility to preview scenes with different algorithms. This is useful if algorithm parameters had to be adjusted.

The scenes were saved in the OGRE-XML file format, because it is supported by many 3D modeling programs like 3dmax through plugins.

To save images, the DirectX 11 API was chosen. The main reason for this choice was to be able to use an existing PCSS implementation of nVidia [19], as well as an existing implementation of the Backprojection algorithm by Michael Schwarz [43]. The gap-filling solution of the Backprojection implementation has been improved by taking mipmap-levels and the angle between light source and surface normal into account (see Section 3.3). Another reason was that the DirectX 11 API provides utility functions for saving textures from the GPU to the hard drive in various formats, like JPG, BMP, or PNG. Apart from this, an implementation in OpenGL also would have been possible.

Images are saved in the BMP image format with a size of $800\text{x}450$ pixels. The BMP format was chosen because it allows lossless storage of pixel data. For rendering the samples twofold multi sampled anti-aliasing (2x MSAA) was used to hide staircase artifacts. This was done because in the pilot study participants tried to find these artifacts in the shadow. MSAA also reduced moiré patterns.

Artifacts that could not be removed this way, affected the umbra and penumbra of the soft shadow. They were not reduced by MSAA, because a wrong form stays wrong even with multi sampling enabled; it is just anti-aliased. Therefore MSAA could be used.

Generation of Votes

A program was written that allows users to compare two images (a trial) with or without a reference image. Participants could specify the difference between the images with a slider interface element and were able to cast votes by pressing a button. After the button had been pressed, the next trial was loaded.

For each vote the participant's ID, the value of the slider, and the ID of the trial were recorded. The participant's ID is an anonymous value that does not provide information about the person's identity. Specific information (age, computer knowledge, experience with computers and computer games) about participants was recorded separately via a questionnaire. The

name of a person was not recorded. The slider value expresses which of the two images of the trial was chosen and lies in the interval $[-1, +1]$. For instance, a slider value of -0.5 in the first block for a trial indicates that the participant thinks the left algorithm is 50% better than right one. This information is removed for evaluation as only three answers are considered ("left preferred","no preference","right preferred").

Environment

The study program ran on two Windows Machines both equipped with the same display (Samsung SyncMaster 226BW with a resolution of 1680x1050) and the default display settings. The program was executed on an empty desktop with a grey background. This ensured that participants were not distracted by a background image. (E.g. during the pilot study a participant asked where he could download the same background image.)

The participants faced the wall and the room was lit by neon lights. Daylight was shut out, because the light situation would not have been controllable. For instance, a cloud could have concealed the sun. Another disadvantage would have been that each participant would have had to be tested at the same daytime.

Evaluation

To evaluate the data generated by the participants, Matlab and R-scrips were written to automate the generation of statistical diagrams and tables.

CHAPTER 5

Analysis

In this chapter we will discuss methods for analyzing the gathered data. The first section will describe how algorithms are compared technically by computing pixel differences between images. This is the way plausibility of soft shadow algorithms is typically measured. We need this technical difference, because we have two images per category and the differences between them have to be comparable. Otherwise we could not use them for our study (see Chapter 4.3). Afterwards we concentrate on analyzing consistence and agreement of users. The consistence is used as an indicator for the quality of votes. This is useful in order to identify participants, whose performance differs significantly from others (see Chapter 6). The next section shows how to evaluate pairwise comparison data and allows us to find algorithm pairs where one algorithm is significantly better than others. As it is desirable to obtain one score to rank the perceptual benefit of each algorithm on a single scale, we will then employ the Bradley-Terry-Luce model to compute these scores.

5.1 Pixel Difference

Because two images need to be generated per category for each algorithm (see Chapter 4), we need to make sure that deviations from the reference image (the ground truth) are similar for both images. Pixel differences of two images $I1$ and $I2$ can be computed by comparing each pixel $\mathbf{I1}_{i,j}$ of image $I1$ to the corresponding pixel $\mathbf{I2}_{i,j}$ in $I2$. The comparison of these vectors is the difference $d(\mathbf{I1}_{i,j}, \mathbf{I2}_{i,j})$ of two pixels. For our purpose we use the *squared euclidean distance* to check if there are any correlations between them and user scores:

$$d(\mathbf{I1}_{i,j}, \mathbf{I2}_{i,j}) = (\mathbf{I1}_{i,j} - \mathbf{I2}_{i,j})^2$$

Other distance functions can also be used, but the squared euclidean distance is more common for this purpose. The sum of all individual differences is the pixel difference:

$$D = \sum_i \sum_j (\mathbf{I1}_{i,j} - \mathbf{I2}_{i,j})^2$$

	PCF	PCSS	BP	GT
PCF	-	1	1	0.5
PCSS	0	-	0	0.5
BP	0	1	-	0
GT	0.5	0.5	1	-

	PCF	PCSS	BP	GT
PCF	-	33	27	15.5
PCSS	45	-	35	29
BP	51	43	-	37.5
GT	62.5	49	40.5	-

Table 5.1: A sample of pairwise comparison matrices A: The left matrix shows a sample of the votes of a single participant in one of the 13 categories. In the right matrix all individual matrices of participants were summed up for a category in a block. (Row algorithms were chosen over column algorithms.)

In statistics, D is called the *mean squared error*. For our purpose we will use $\sqrt{D}$, the *root mean squared error*, because it is more commonly used. A value of 0 indicates that there is no difference between the two images and 1 that the images are completely different (e.g. one is black the other white).

There remains a problem with using all pixels of an image. Shadows affect only parts of an image and results become higher the more shadow is visible. Since we want to compare shadows to each other, we only consider pixels that lie inside a soft shadow (see Figure 4.5). This also allows us compare shadow deviations across images and we can ensure that pixel differences are similar for all images.

5.2 Consistence, Agreement, and Indifference

As proposed in a similar pairwise comparison-study by Setyawan and Lagerdijk [46], it is useful to analyze the *coefficient of consistence* and the *coefficient of agreement*, which are sensitive if *one* user does not consistently choose one algorithm over another, or if *all* users do not consistently choose one algorithm over another respectively. In the rest of the document we will refer to these coefficients simply as *consistence* and *agreement*. We combine these two measures to show that learning occurred after Block II (see Chapter 6). We additionally introduce a *degree of indifference* which lets us find scenes that are very similar. This is either because a scene provides too few cues for a user to make a decision or because of a high amount of noise (e.g., users saw differences, but could not decide if they are errors or not).

To be able to compute these measures, we derived a 4×4 matrix A of pairwise comparisons (see Table 5.1). The size of the matrix is defined by the number of algorithms $t = 4$. In each cell A_{ij}, where $i \neq j$, we have an entry specifying the number of preferences of algorithm i over j. For a category matrix, this is the number of participants that thought i looks better or more plausible than j. The same is true for A_{ji} only that j is chosen over i and not vice versa. But participants can also decide to choose none of the two algorithms. The amount of these "no-preference" responses A^0_{ij} can be obtained by subtracting A_{ij} and A_{ji} from the number of participants ($A^0_{ij} = N - A_{ij} - A_{ji}, i > j$).

Since *consistence* and *agreement* assume no neutral responses, we distribute no-preference responses by dividing A^0_{ij} by 2 and adding each half to the preference counts in A_{ij} and A_{ji}. This strategy assumes that a participant without preference, would choose i over j in 50% of the

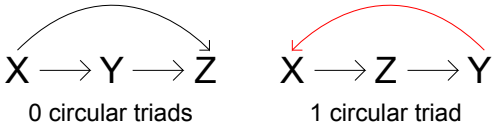

Figure 5.1: The left sample shows a response without contradictions. No circular triads occur as Y is better that X, Z better than Y, and Z better that X. In the right sample there is one contradiction that results in one circular triad.

cases. There is a matrix for each participant in each of the 13 categories in each block, resulting in $N \times 13 \times 4$ matrices (see left matrix in Table 5.1). These matrices are needed to compute the consistence. For the agreement we need to calculate the sum per category (13×4 matrices). The right matrix in Table 5.1 is an example of such a matrix.

Each matrix is filled with 12 values ($t^2 = 4 \times 4 = 16$ cells minus 4 empty cells where $i = j$). Because A_{ij} and A_{ji} result from the same comparison, there are 6 possible comparisons between the 4 algorithms (PCF-PCSS, PCF-BP, PCF-GT, PCSS-BP, PCSS-GT, BP-GT). We need these definitions in the following sections.

Coefficient of Consistence (ξ)

The coefficient of consistence ξ is a measure to score the amount of contradictory responses and is used to judge the quality of votes (e.g. if a participant took the study seriously). It is computed by counting or estimating the amount of so called *circular triads*, which occur when algorithm X is preferred over Y, Y over Z and X over Z (see Figure 5.1). This is an inconsistency. If Z is better than Y and Y is better than X then Z should be better than X and not worse. In order to compute the coefficient of consistence we need to count the number of circular triads. For larger t this can become quite time consuming. Luckily ξ can be computed as follows [28]:

$$c = \frac{t(t^2-1)}{24} - \frac{\sum_i (a_i - \bar{a})^2}{2} \tag{5.1}$$

In this equation, t corresponds to the number of columns in A ($t = 4$ algorithms in our case), and the values a_i correspond to the sum of the scores of row i and $\bar{a}$ to the average score over all a_i. We generate pairwise comparison matrices for each category and each participant. Since we accommodate ties by dividing the number of no-preference responses A^0_{ij} by 2 and adding each half to the preference counts in A_{ij} and A_{ji}, a cell in the matrix can either be 0, 1, or 0.5. If all cells contain 0.5, all a_i are the same (1.5) and therefore equal to $\bar{a}$. In this case the term to the right of the minus operator becomes 0 which maximizes c to $t(t^2-1)/24$. c can be minimized, if 0 and 1 are only present at one side of the matrix. In this case the term to the right of the minus operator becomes $t(t^2-1)/24$ and $c = 0$.

Having an estimate with accommodated ties, we obtain the coefficient of consistence ξ by:

$$\xi = 1 - \frac{24c}{t(t^2-1)} \tag{5.2}$$

In equation 5.2, $24c/t(t^2-1)$ normalizes c to the range $[0,1]$. This part, on the right side of the minus operator, is a measure of inconsistence, but since we want to measure consistence, the normalized part needs to be flipped. Now consistence ξ reaches a maximum of 1 when there are no circular triads, and a minimum of 0 when the maximum number of circular triads is reached. 0 is expected for random responses as mentioned in the description of Equation 5.1.

Note that until now we counted the number of circular triads for each participant in each category individually. To evaluate consistency per block, category, or person, we have to calculate the mean of the individual results accordingly.

The coefficient of consistence is particularly useful to evaluate the task performance of particular participants in the study and identify those with low consistence, which should be excluded from the analysis (see participant 43 in Figure 6.1). It can also be used as an indication of the similarity of the algorithms in a scene or block (see Figures 6.5 and 6.3 respectively).

Coefficient of Agreement (u)

To compute the coefficient of agreement u, it is necessary to sum up the number of participant pairs which agree in their choice (τ). This is, because maximal agreement is reached if all participants agree in their choices (one side of the pairwise comparison matrix contains only 0 and the other only N for N participants). If all cells of the matrix contain values of $N/2$, minimal agreement is reached. Each time 2 participants make the same decision regarding a pair of algorithms, there is one agreement for this pair. The number of pairs for one comparison A_{ij} is $\binom{A_{ij}}{2}$. For all algorithm comparisons we measure the agreement by counting the number of participant pairs that make the same decision [46]:

$$\tau = \sum_i \sum_j \binom{A_{ij}}{2} \tag{5.3}$$

Each participant can potentially agree with each other participant so the maximum of possible participant pairs is $\binom{N}{2}$. As there are $\binom{t}{2}$ algorithm combinations, the maximum τ is $\binom{t}{2}\binom{N}{2}$. According to Kendall/Babington-Smith [28], the coefficient of agreement u is then defined as follows:

$$u = \frac{2\tau}{max(\tau)} - 1 = \frac{2\tau}{\binom{t}{2}\binom{N}{2}} - 1 \tag{5.4}$$

The equation transforms τ into the interval $[-1/(N-1), 1]$ for even and $[-1/N, 1]$ for odd N. We try to illustrate the principle of agreement between two algorithms with an example: When algorithms i and j are compared by 2 participants, the probability that both agree in their preference depends on the actual contrast between i and j, and the precision of the participants' perception. As illustrated in Figure 5.2, agreement decreases with the overlap in the distributions of participants' responses (e.g., perceived quality) on i and j, which can be either due to a low precision (i.e., high random error) or a high similarity of i and j.

To accommodate ties, the same strategy, used to compute the coefficient of consistence in Chapter 5.2, is employed. However, this strategy introduces the possibility to obtain non-integer entries for A_{ij}. Hence we use the computationally less expensive continuous approximation of

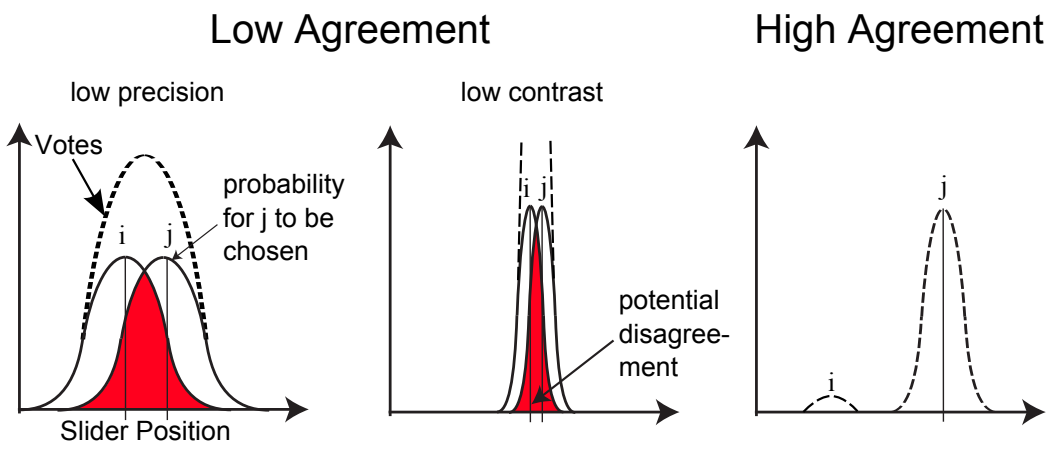

Figure 5.2: An illustration of low and high agreement. Low agreement can occur, if participants are not sure whether algorithm i or j is better, or if the algorithms produce very similar results. In this case the slider responses for i and j will overlap which results in low agreement.

the binomial coefficient, which also accepts floating-point values:

$$\tau = \sum_i \sum_j \binom{A_{ij}}{2} \approx 0.5 \left(\sum_{i \neq j} A_{ij}^2 - N \binom{t}{2} \right) \tag{5.5}$$

This coefficient can reach a maximum of 1.0 when there is total agreement (e.g., each entry in A_{ij} is either 0 or N as in Table 5.3). The minimum is -1.0, which occurs when $N = 2$ and each A_{ij} contains 1. For $N > 2$ it is $-1/(N-1)$ for even and $-1/N$ for odd N. A sample for minimal agreement with $N = 48$ is shown in Table 5.3. The rules of thumb for u are: < 0.0 no agreement, $[0.0, 0.2)$ slight agreement, $[0.2, 0.4)$ fair agreement, $[0.4, 0.6)$ moderate and > 0.6 high agreement.

Degree of Indifference (ι)

Since it is not possible to decide whether low agreement is due to a low precision of the measurement method, or due to a high similarity in the algorithms being measured, we analyze the amount of no-preference responses A_{ij}^0.

We define a measure that gives us a good hint about the degree of similarity of algorithms being compared, which we denote as *indifference* ι. We compute it from the proportion of no-preference responses A_{ij}^0 among all (N) comparisons being performed. We normalize this score with the expected maximum of no-preference answers, which we obtain empirically from the no-preference responses A_{ii}^0 in the N_0 comparisons where the participants were shown pairs of identical images:

	PCF	PCSS	BP	GT
PCF	-	0	48	0
PCSS	48	-	0	48
BP	0	48	-	0
GT	48	0	48	-

Table 5.2: A sample of a pairwise comparison matrix with a maximal coefficient of agreement ($u = 1$).

	PCF	PCSS	BP	GT
PCF	-	24	24	24
PCSS	24	-	24	24
BP	24	24	-	24
GT	24	24	24	-

Table 5.3: A sample of a pairwise comparison matrix with a minimal coefficient of agreement ($u = -0.0213$).

$$\iota = \frac{\frac{1}{N}\sum_i \sum_{j>i} A^0_{ij}}{\frac{1}{N_0}\sum_i A^0_{ii}} \tag{5.6}$$

5.3 Analyzing Results of Pairwise Comparisons

Participants respond with a continuous slider, which has a snapping to the center to indicate zero preference. For further analysis we transform this data into the three nominal categories "left preferred", "no preference", and "right preferred", which were used for further analysis.

To find out which algorithms perform better than others, we define that algorithm j is superior to i when the majority of users prefers j over i. So for N participants we have to find out how many of them need to vote for algorithm j in order for j to be significantly better than i. In other words we want to find a threshold k_α, for which all $j \geq k_\alpha$ are significantly better than i for a confidence interval of $\alpha = 0.05$. k_α can be computed with the Cumulative Binomial Distribution F^N_p which computes the probability for i to be chosen more often (or less) than k times over j. This probability can be seen as a confidence interval. For instance, if we wanted to know how high the chances are that algorithm i is chosen less than k times over j, we need to sum the binomial probabilities for i to be chosen 0, 1, 2, ..., or $k - 1$ times:

$$b(x < k) = b(x = 0) + b(x = 1) + ... + b(x = k - 1)$$

If we want to find out how often i has to be chosen over j so i is significantly better, we need to search for an integer k_α, where the Cumulative Binomial Distribution $b(x \geq k_\alpha)$ is just less than the minimal tolerated probability α (the *confidence interval*). For $N = 48$, $\alpha = 0.025$, and probability $p = 0.5$ for an algorithm to be selected, we find that a minimum of $k_\alpha = 32$ participants have to vote for i ($b(x \geq 32)$). To find out how many votes are needed for j to be significantly better, we need to search for an integer k_α for which the Cumulative Binomial Distribution is less than $\alpha = 0.025$ (which is $b(x \leq 16)$).

Computing k_α we can now easily find significances by plotting the probabilities for the 6 algorithm comparisons $C_{i,j}$ in combination with *significance-lines* (see Figure 6.4). An algorithm is significantly better than the other if it exceeds the farthest removed significance-line. I.e., if upper bars lie below lower red lines or lower bars above upper red lines.

5.4 Obtaining Scores from Pairwise Results

In Chapter 5.3 we discussed how to compare two algorithms to each other. This way we can easily find out if one algorithm is better than another, but we have no idea about how all these individual results affect the overall rating of the algorithms. It is desirable to obtain one score to rank the perceptual benefit of each algorithm on a single scale, so we can say how the algorithms performed against each other.

The Bradley-Terry-Luce model [8], was developed to obtain such ratings. The model uses pairwise comparison data from so called *latent variables* (e.g. taste or plausibility) to calculate ratings π_i. The original Bradley-Terry-Luce model was designed to evaluate forced choice (two choices) data. Fortunately there is a proper extension of this model available that accommodates ties [15]. Pietsch gave a step by step introduction on how to use the Bradley-Terry-Luce model to evaluate pairwise comparison data with ties in [36], which we adapted for our work.

In this model the probabilities for a comparison $C_{i,j}$ to result in one of the three outcomes k (1: "i preferred", 2: "j preferred", and 3: "no preference") are:

$$\begin{gathered} p_{ij1} = \frac{\pi_i}{\Pi} \quad p_{ij2} = \frac{\pi_j}{\Pi} \quad p_{ij3} = \frac{v\sqrt{\pi_i \pi_j}}{\Pi} \\ \Pi = \pi_i + \pi_j + v\sqrt{\pi_i \pi_j} \end{gathered} \tag{5.7}$$

where π_i is the real probability of the i^{th} treatment (algorithm) to occur and v is a constant, defining the probability for differences ($v\sqrt{\pi_i \pi_j}$ is so to say π_k).

The fitting of the Bradley-Terry-Luce model is done with a Generalized Linear Model (GLM) which has the basic form:

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\theta}$$

$g()$ is the *link-function* and $\boldsymbol{\mu}$ the vector of mean values $\mathbf{Y}$ of the observations ($n_{ij}p_{ijk}$). $\mathbf{X}$ is the model matrix and $\boldsymbol{\theta}$ the vector of unknown parameters that needs to be computed. For binomial distributions $\boldsymbol{\theta}$ has the form $(\theta_1, \theta_2, ..., \theta_{t-1}, 0)$. For trinomial distributions $\mathbf{Y}$ is a vector of independent trinomial-distributed stochastic variables. Such distributions cannot directly be implemented as a GLM. According to Fienberg et al. [21], $\mathbf{Y}$ has to be modeled as a vector of independent Poisson-distributed stochastic variables for a GLM implementation. To achieve this, we have to extend the parameter vector $\boldsymbol{\theta}$ by $log(v)$ and the $t(t-1)/2$ normalization constants D_{ij} as described in [12, 36]. $\boldsymbol{\theta}$ has now the following form:

$$\boldsymbol{\theta} = (\theta_1, ..., \theta_{t-1}, \theta_t, log(v), D_{12}, ..., D_{t-1,t}) \tag{5.8}$$

$\theta_1, ..., \theta_{t-1}$ contain the ratings of our three algorithms compared to the ground truth. Because we use Poisson-distribution, the link-function $g()$ has to be $log()$. We therefore need the alternative (but equivalent) representation for Equations 5.7 that includes the normalization constants D_{ij}:

$$\begin{gathered} log(n_{ij}p_{ij1}) = \theta_i + D_{ij} \quad log(n_{ij}p_{ij2}) = \theta_j + D_{ij} \\ log(n_{ij}p_{ij3}) = log(v) + 0.5(\theta_i + \theta_j) + D_{ij} \end{gathered}$$

where $\theta_i = log(\pi_i)$. n_{ij} is the number of comparisons between i and j, and $n_{ij}p_{ijk}$ is the mean value of "i preferred over j".

Now the model matrix $\mathbf{X}$ needs to be defined in order to complete Equation 5.8. The matrix has $3t(t-1)/2$ rows (3 outcomes and $t(t-1)/2 = 6$ comparisons) and $t + 1 + t(t-1)/2$ columns (see Equation 5.8) and is created by the following steps:

1. For the (i;j;k)-row with k=1 the m^{th} column is 1, if m=i.
2. For the (i;j;k)-row with k=2 the m^{th} column is 1, if m=j.
3. For the (i;j;k)-row with k=3 the m^{th} column is 0.5, if m=i or m=j. It is 1 if m=t+1.
4. For the (i;j;k)-row the m^{th} column is 1, if $m = t + 1 + j$.
5. Fill the rest with 0.

The linear fitting is performed such that the ground truth is per definition the zero baseline. Note that the matrix is not of full rank and we have to restrict parameter θ_t of vector $\boldsymbol{\theta}$ to 0 and the t^{th} column of $\mathbf{X}$ must be excluded [36]. The results of the fitting are the parameters θ_i corresponding to the ratings on a log scale ($\pi_i = exp(\theta_i)$) for each algorithm i (see Equation 5.10).

In Figure 6.2 the results obtained by fitting the model to the data pooled over all scenes are plotted. On the right hand side of this figure we visualize the ratings as proportions, because portions are more intuitive to compare to each other. We get portions by transforming θ_i to π_i and using the fact that $\theta_i = log(\pi_i) \Leftrightarrow pi_i = exp(\theta_i)$. p_i can then be computed as follows:

$$p_i = \frac{exp(\theta_i)}{\sum_k exp(\theta_k)} \tag{5.9}$$

With the resulting statistics (Figures 6.7, 6.8, and 6.4) it is easy to see how algorithms perform compared to the ground truth and each other. We also see how the four algorithms perform in each category and in each block. We use this knowledge to find out which algorithms are comparable and which are superior to others (see Chapter 6).

$$
\begin{pmatrix}
\theta_1 + D_{12} \\
\theta_2 + D_{12} \\
log(v) + 0.5(\theta_1 + \theta_2) + D_{12} \\
\theta_1 + D_{13} \\
\theta_3 + D_{13} \\
log(v) + 0.5(\theta_1 + \theta_3) + D_{13} \\
\theta_1 + D_{14} \\
D_{14} \\
log(v) + 0.5\theta_1 + D_{14} \\
\theta_2 + D_{23} \\
\theta_3 + D_{23} \\
log(v) + 0.5(\theta_2 + \theta_3) + D_{23} \\
\theta_2 + D_{24} \\
D_{24} \\
log(v) + 0.5\theta_2 + D_{24} \\
\theta_3 + D_{34} \\
D_{34} \\
log(v) + 0.5\theta_3 + D_{34}
\end{pmatrix}^T
=
\left|
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0.5 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0.5 & 0 & 0.5 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0.5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0.5 & 0.5 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0.5 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0.5 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
\right|
\begin{pmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
log(v) \\
D_{12} \\
D_{13} \\
D_{14} \\
D_{23} \\
D_{24} \\
D_{34}
\end{pmatrix}
\tag{5.10}
$$

CHAPTER 6

Results

We conducted an experiment with 48 participants (age mean = 26.27, stdev = 7.57, 25 female, 23 male) which were recruited by advertising in university related online forums. The participants were randomly assigned to one of the two groups (Group A: male = 12, female = 13; Group B: male = 11, female = 12). The experiment took on average 55 minutes per person.

The participants cast 13104 votes, which are evaluated statistically in this chapter. We compute the *coefficient of consistence* for each participant to get a value for the quality of the votes. This allows us to find participants that do not have enough consistency to be considered valid. We will also compare the results of *consistence*, *agreement*, *indifference*, and *pairwise comparison* to each other to find out, if we can simulate *inexperienced*, *experienced*, and *expert users* with our block design.

To get a first insight of the data, results were pooled per block to see which algorithms perform generally better than others. Differences between user groups are also measurable this way. We then show how these results can be used to select the right algorithm for specific user groups.

The final evaluation is done per category to see which categories provide cues for users to judge soft shadows. Evaluating the results per category shows us for which categories an algorithm is suited.

6.1 User and Study Design Analysis

In order to remove data of participants that did not take the experiment seriously, we computed the *coefficient of consistence*. Looking at all participants separately, as shown in Figure 6.1, we have balanced scores over all participants except one (43), who was removed from the data. This shows that the performance in the perception task does not vary much between participants. It also shows that there are no distinct clusters where some subjects perform much better or worse than others.

Consistence is also on the better side for each block (Figure 6.3) and each category (Figure 6.5). A small increase of consistence can be observed after Block II in Figure 6.3. Though it is

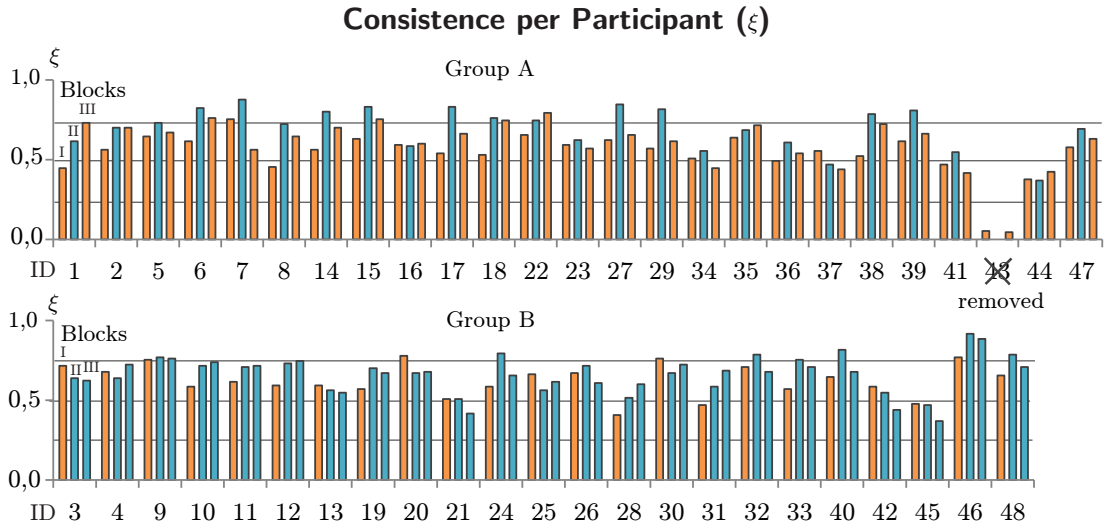

Figure 6.1: Consistence values per participant. There is one participant where consistence is very low compared to the rest. We therefore have to expect invalid data from this person.

minimal, it indicates that participants are better able to judge soft shadow algorithms after Block II, which is a sign that our block design works. We can also see an increase of *agreement* after the second block, which means that more participants voted similar in Block III than in Block I. These votes are visualized in Figure 6.6. After Block II the physically more plausible algorithms received more votes than in Block I (the lower bars grow while the upper bars shrink). With our previous observations we can say that subjects got a better understanding of soft shadows after Block II. So in Block III we indeed have *experienced* users and the block design works.

6.2 Data Pooled Over Categories

To get a first insight about the perceived quality of the soft shadow algorithms, we pooled the data over all categories for each block and Groups A and B in Block III. The pooled results of the pairwise comparisons are depicted in Figure 6.4. We determine significant results by drawing red lines corresponding to the 95% confidence intervals. Preference scores lower or equal to these margins are significant if upper bars lie below the lower red line or lower bars above the upper red line.

There is no significant preference of one algorithm over another in Block I, while differences become clearer for Block II, where the ground truth was provided as reference. Nevertheless, participants were not able to distinguish the ground truth from images generated with Back-projection or PCSS. In Block III results are more distinct than in the first block, but similar to Block II, additionally proving that learning has occurred and that our block design works.

Looking at the pooled data in all Blocks (Figure 6.4), the only significant difference is found in comparison to PCF, the lowest-quality soft shadow. It is the only algorithm in our study that

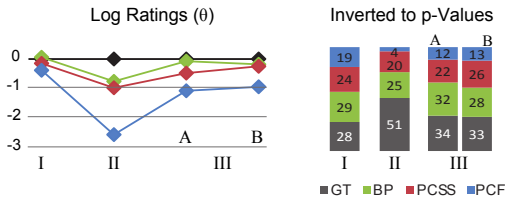

Figure 6.2: Overview on log-ratings and p-values for data pooled over all categories.

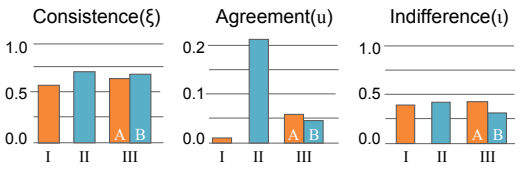

Figure 6.3: Overview on consistence, agreement, and indifference for data pooled over all categories.

cannot handle *contact hard soft shadows*. In comparison PCSS provides a crude approximation of this feature, but is already superior to PCF for *experienced* and *expert users*. Moreover, PCSS is already plausible enough so Backprojection and ground truth do not perform significantly better. We can conclude that for the general user (who lies somewhere between *inexperienced* and *expert users*) contact hard soft shadows are needed to provide plausibility, but a correct estimation of the penumbra is not necessary.

For a specific user group the statistics of Figure 6.4 can be used to find the preferred soft shadow algorithm in terms of performance and/or plausibility. E.g., the statistic of the inexperienced user (Block I) shows that no algorithm is significantly better than PCF, the fastest algorithm. But the ground truth and Backprojection come close to the red line and are therefore *nearly* significantly better. For PCSS, the second fastest algorithm, neither ground truth nor Backprojection come close to the upper red line. So if an application is performance limited we choose PCF, because it is just good enough for inexperienced users and we have more time for other computations. Otherwise, if plausibility is critical, we choose PCSS.

For a group of experienced users (Block III Group A and B) Figure 6.4 shows that PCSS provides both, plausibility and performance, and is the preferred algorithm for this group. For expert users (Block II) we see that the ground truth is nearly significantly better than PCSS, resulting in a similar situation as in Block I. So we choose Backprojection, if the application should be believable and PCSS if we are concerned about performance. PCF is no option as all other algorithms are significantly or, in the case of Block III Group A, almost significantly better.

Overall there are small differences between algorithms apart from PCF, when looking at the ratings shown in Figure 6.2. A sufficiently clear difference can only be seen in the ratings obtained for Block II. It is also the only block where agreement is on an acceptable level (see Figure 6.3). Looking at indifference scores, it seems that there is no change between blocks, which is reasonable since the same categories and algorithms were evaluated in all blocks. The only difference is the measurement method (with or without ground truth reference). We conclude that low agreements in the other blocks are subject to latent factors (e.g., the pooled categories), which produce a higher amount of random noise when the ground truth reference is not available.

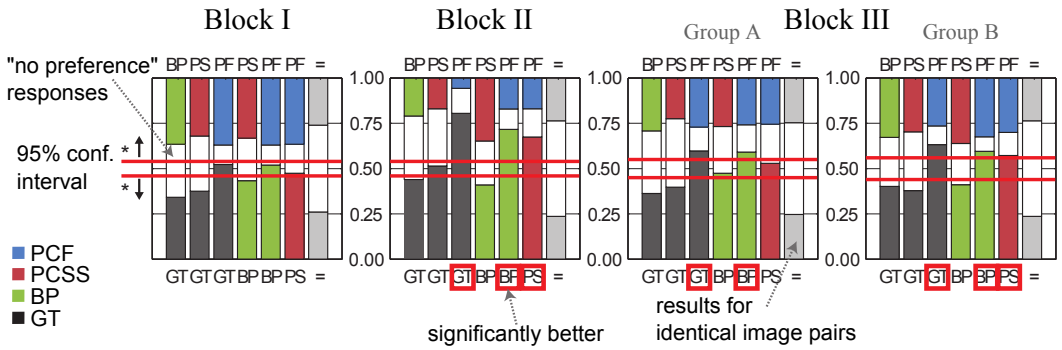

Figure 6.4: Visualization of pairwise comparison results for each block. An algorithm is significantly better than the other if it exceeds the farthest removed red line (confidence interval of 95%). E.g., if upper bars lie below lower red lines or lower bars above upper red lines.

6.3 Data Separated by Category

Since differences of the results are clearer in some scene categories, it is useful to regard the results per category. In Figure 6.5, we list all results for consistence, agreement, and indifference by each scene category separately. To interpret the results for the ratings, it is important to look at the level of agreement and indifference simultaneously. If agreement is low and indifference high as in Category 1, scenes provide few cues that allow users to distinguish shadows. In other words, the perceived difference in the shadow quality is low. Looking at ratings in Figure 6.7, we can see that such scenes do not provide much information and can therefore be skipped in future studies. Relatively low agreement combined with low indifference indicates that the difference is low due to a high amount of noise (e.g., random errors by latent factors). From the users' point of view, when asking about realism, this means that they saw differences, but could not decide if they are errors or not. When asking about the better looking soft shadow it indicates that it was hard for users to decide what makes a soft shadow "better looking". Categories 9, 11, 12, and 13 show such combinations.

For categories 2, 11, and 12 agreement is high and indifference low for some blocks. This indicates that these three categories provide soft shadows that differ strongly from algorithm to algorithm so users can better distinguish between them and have a clear preference to specific algorithms.

For Category 11, Figure 6.5 shows that agreement drops after Block II, meaning that people had a wrong idea of the soft shadow in Block I. We can prove this by looking at the pairwise comparison results in Figure 6.6 and at ratings in Figures 6.7 and 6.8. As can be seen in Figure 6.8, in Block I only 6% chose the ground truth solution. In Block III Group B 35% recognize the physically most plausible algorithm. In Block III Group A however, still only 15% regard the ground truth as *aesthetically* more pleasing. The ratings for Category 12 show a similar behavior, with Backprojection scoring over 50% in both scenes for Block I, and also PCSS scoring much higher than the ground truth. Some participants told us that they preferred the "darker" looking shadows in these categories. Others said that they chose Backprojection, because the

shadow was “sharper”. The gap-filling approach caused visible overshadowing in this scene, which resulted in more pronounced and sharper soft shadows. The reason why users preferred sharper and/or darker soft shadows is not apparent from our data. It is possible that gap-filling artifacts resembled shadows in real world scenarios and were hence considered better looking. It is also possible that the random nature of the vegetation scenes had an impact on votes. This has to be clarified in future works.

6.4 Discussion

In general, analyzing consistence, agreement, and indifference is helpful during pilot studies, where categories should meet specific conditions. For our study we wanted to have representative categories to find out how different soft shadow algorithms perform. Too many categories behaving the same way indicate either that we did not select a representative set of algorithms, or that some categories we considered different are actually the same (are of the same category). Our results show that we have selected categories ranging from low consistence with low agreement and high indifference, to high consistence with high agreement and low indifference, and to everywhere in between. So the set of scenes is representative for our study.

As can be seen in Figure 6.7, a clear difference between the perceived quality of the algorithms can only be observed in Block II, where the reference solution is visible. In blocks without reference solution, the differences between algorithms are usually very small. Interestingly, averaged over all scenes, the final ranking of the algorithms corresponds to how physically based they are (PCF $<$ PCSS $<$ BP $<$ GT). While this result is of course most pronounced for Block II, it nevertheless holds true over all blocks, which was not to be expected a priori. It indicates that people have at least some intuition of how a physically correct shadow should look like and prefer plausible soft shadows. It also indicates that a comparison with a reference solution (as it is used in rendering papers) is a valid and useful method to assess the quality of a shadow algorithm. Not only because it gives the clearest results, but also because the results can be legitimately extrapolated to the case of both, trained and untrained users, without reference solution.

On a per-scene level, the ranking can change a lot between categories, and there is no clear winning algorithm which is consistently better than the others. With what we have learned from our study, we can single out several facts about each real-time soft shadow algorithm:

PCF In Block I PCF is not much behind the other algorithms, and can be seen as a viable method for faking soft shadows that are convincing for the casual user. For example, according to Figure 6.4, the probability that users choose the ground truth over PCF is only 9% higher. Hence, PCF is sufficient for shadows on less powerful devices, like mobile phones, where the smaller screen further reduces shadow perception, or for outdoor scenes with low penumbra variation. A clear learning effect can be observed between Block I and Block III in terms of discerning PCF from the ground solution (the probability of choosing PCF drops from 19% to under 13% in Block III). Therefore, high-profile games, which strive for realism and attract experienced gamers, who presumably already fall into the

category of trained users, require more sophisticated shadowing methods that with contact hardening soft shadows.

PCSS Once large penumbra variations come into play, the additional cost of PCSS pays off (as can be seen in Figure 6.7, Category 2). Regarding PCSS versus Backprojection, the overall scores of these two algorithms are very close, with just a slight advantage for Backprojection. Per scene, PCSS consistently scores as good as, or slightly worse than Backprojection (with the exception of the vegetation scenes, which have to be treated with caution as discussed above). This result makes it hard to justify trading the speed and robustness of PCSS for the slightly better perceived shadow quality of Backprojection, especially given the fact that in practice, it is hard to avoid artifacts caused by Backprojection (as we have done in these artificial tests). Besides *gap-filling*, there are other approaches to avoid artifacts [42, 44], which are more robust and easier to adjust but more complex to implement.

Backprojection As can be seen in Figure 6.7, Backprojection behaves very similar to the ground truth solution over all blocks (except for Categories 11 and 12 as we discussed previously). Hence we conclude, as long as the artifacts can be handled well, there is no need to go beyond Backprojection in terms of physical plausibility in practical cases. The results and experiences from our pilot studies imply that handling artifacts and increasing robustness is more important and has to be investigated in future works.

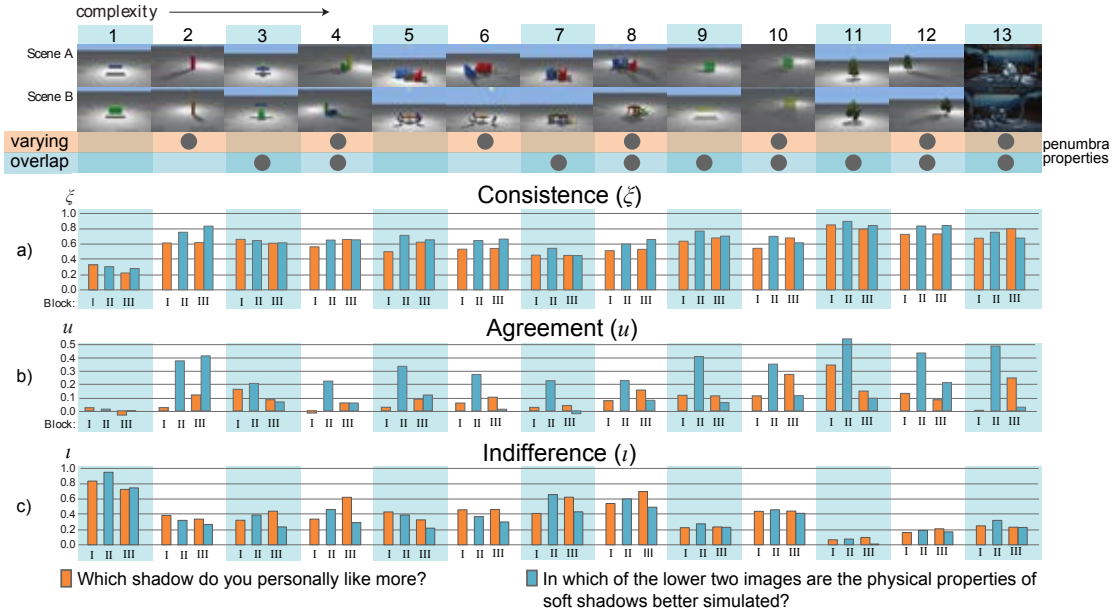

Figure 6.5: Coefficient of consistence, coefficient of agreement, and degree of indifference for all scene categories and blocks.

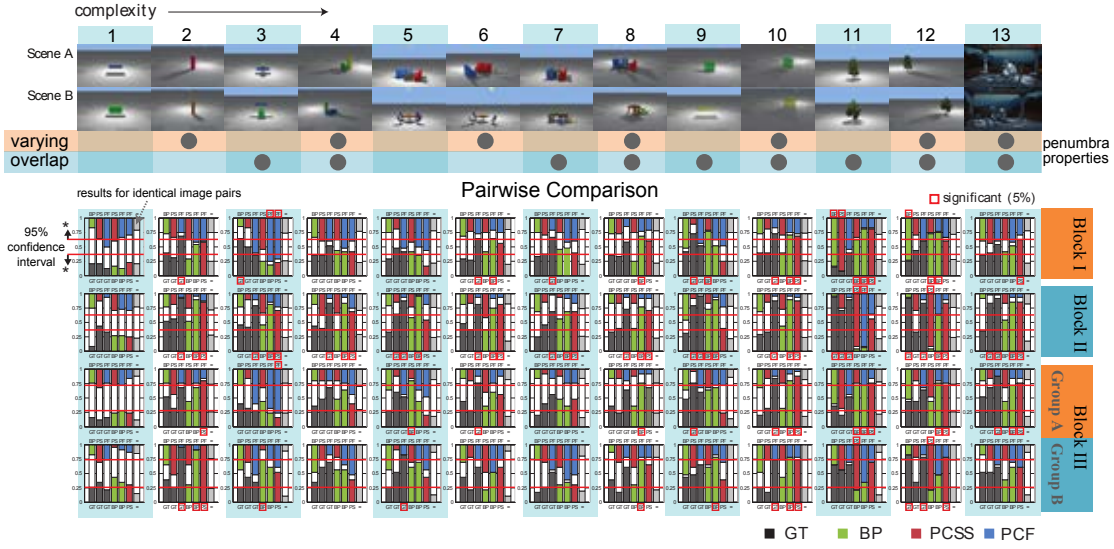

Figure 6.6: Pairwise comparison results for all scene categories and blocks.

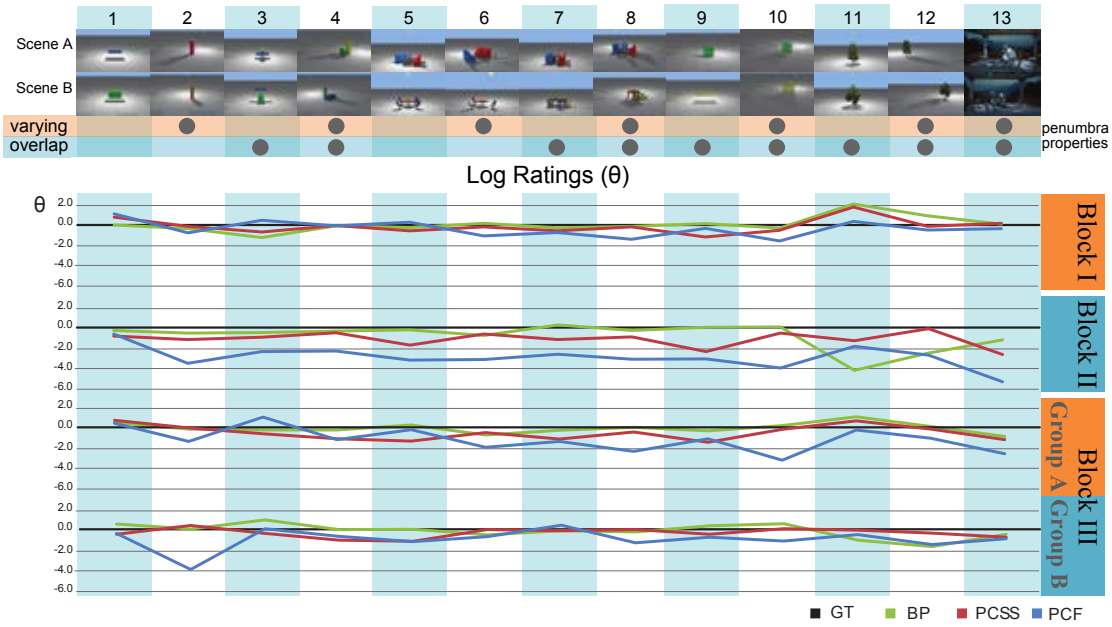

Figure 6.7: Log-ratings for all scene categories and blocks.

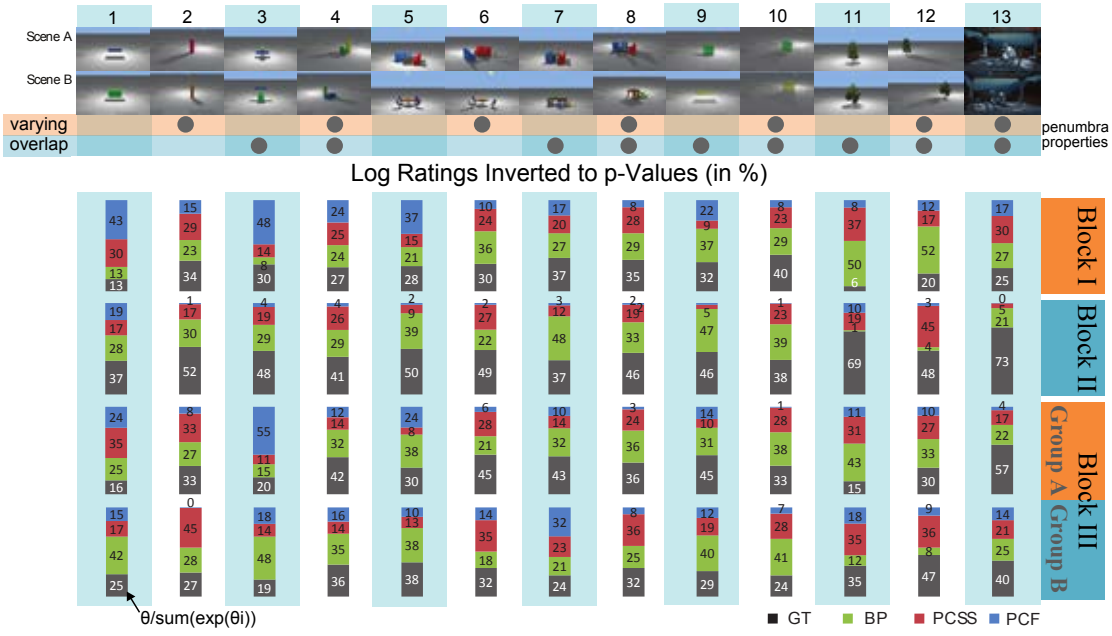

Figure 6.8: Log-ratings inverted to p-values for all scene categories and blocks.

CHAPTER 7

Conclusion

In this study we evaluated the sensitivity of human perception regarding properties of soft shadows. We focused on physical plausibility and aesthetic aspects. Our study shows several interesting results, which are useful for future shadow research and provides a scientific prove for intuitions experts already have about soft shadows (e.g., that the ability of a user to judge soft shadows without reference is low). An interesting finding of this study is that, on average, the results of a comparison using the reference solution correlates well with a comparison without reference solution and is representative for the standard gaming or movie experience case. As a practical contribution, this study provides guidelines that can be used to avoid costly methods if they do not pay off in a particular real-time application (as a cheaper method may even be considered more plausible in some scenarios). We could show that approximating simple contact hardening in soft shadows (as in PCSS) is sufficient for the "average" user and not significantly worse for experts. We could also show that going beyond the physical plausibility of the Back-projection algorithm is not reasonable. The work on future soft shadow algorithms has to focus on robustness or increasing performance by simplified penumbra estimation. The results of this work allow us to better judge the plausibility of existing and future shadow algorithms for inexperienced, experienced, and expert users.

We also see this study as a starting point to exploring the complex, multi-dimensional space that constitutes all aspects of soft shadows. In particular the statistical methodology used to compare and rank multiple algorithms using pairwise comparisons without resorting to the 2-option forced choice paradigm. The study design is of particular interest to research that tries to evaluate different levels of experiences, without laboriously searching for the right candidates. Using this framework as a basis, we can think of several aspects which were left out in this study but are good candidates for future work. E.g., rendering artifacts in soft shadows, impact of animation, and do users prefer sharper and/or darker shadows in terms of aesthetics. Also, we would like to investigate how sensitive the human visual system is towards potential inconsistencies that arise, if the appearance of the projected shadow does not match the real geometry. We believe that such questions are important for practical applications. For example, while the

movie industry is not interested in physical correctness of the illumination, there is a question of how far the solution can deviate from the ground truth and be considered believable.

While we strove to examine all possible factors which constitute the worst case, there is a limit to the dimensionality which can be handled in sufficient detail within the scope of a single study. Animation, for instance, is an important factor which allows us to deduce information about the soft shadow shape, as it gives cues to estimate the relations between caster, receiver, and light source. It can also reduce the same cues if changes happen too fast. Hence, in future work we want to incorporate animation of the camera, the light source, and the scene objects as parameters. Furthermore, we want to provide in-depth analysis of the effects of (high-frequency) noise and textures on shadow perception.

APPENDIX A

Additional Figures

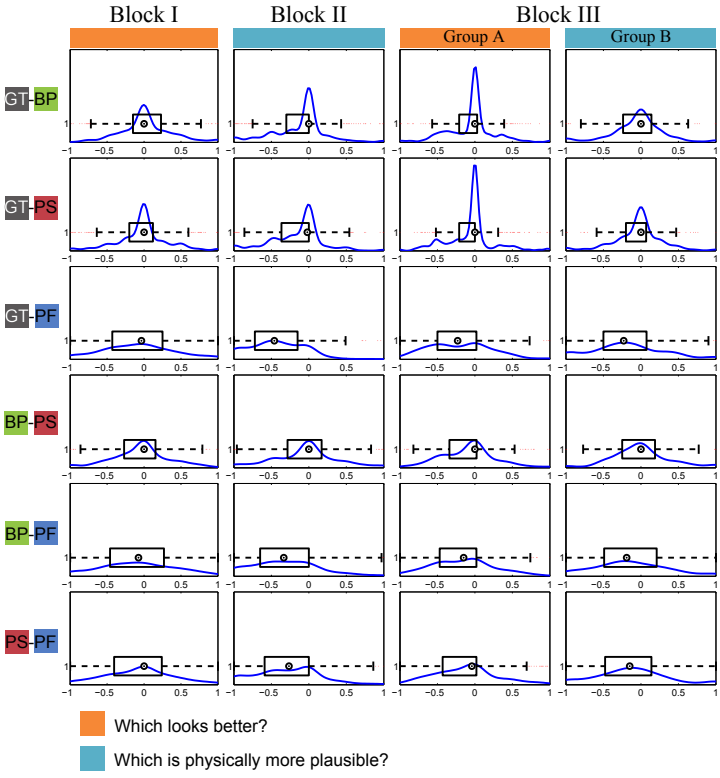

Figure A.1: The distribution of user votes pooled over all categories. The blue lines were generated by a *kernel smoothing density estimation* to display where participants placed votes. The boxplots provide a statistical visualization.

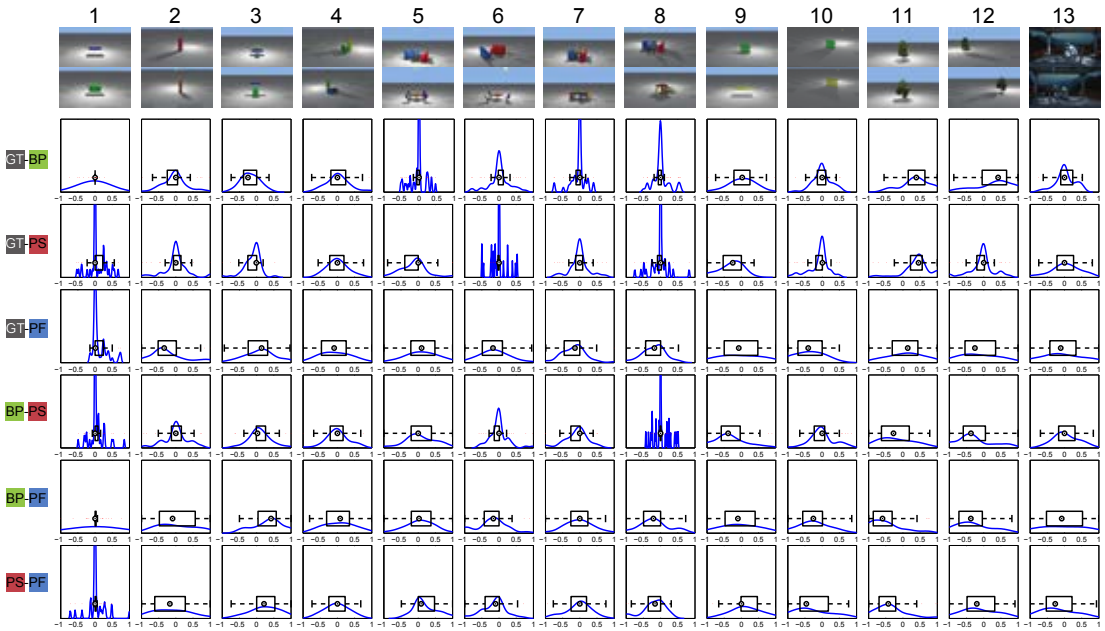

Figure A.2: Slider distribution in Block I (“Which shadow looks better?”).

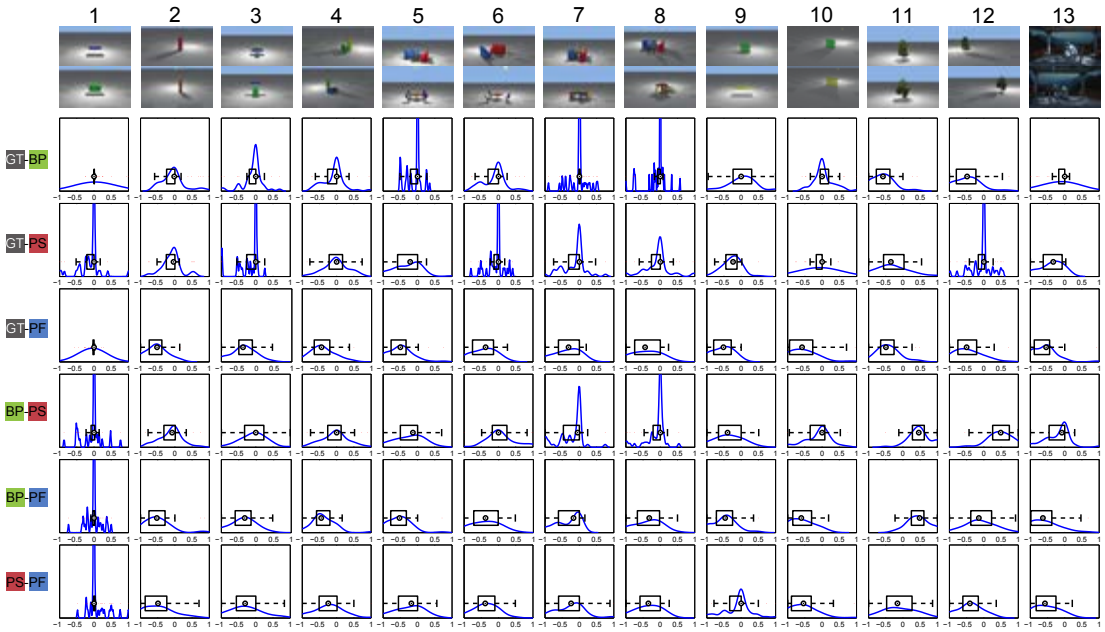

Figure A.3: Slider distribution in Block II (“Which is physically more plausible?”).

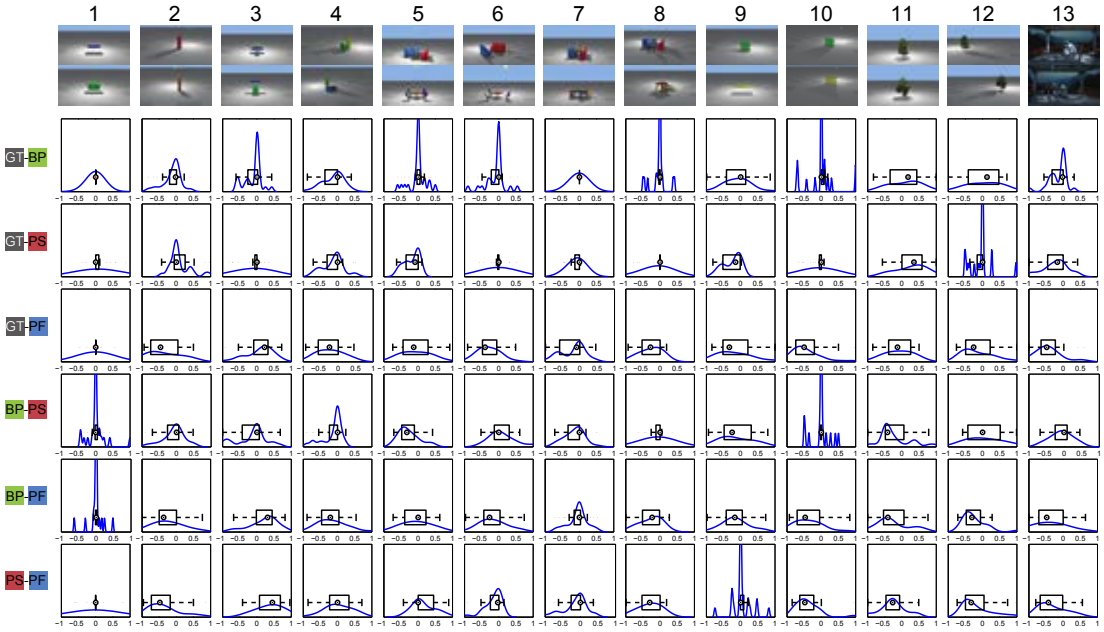

Figure A.4: Slider distribution in Block III Group A (“Which shadow looks better?”).

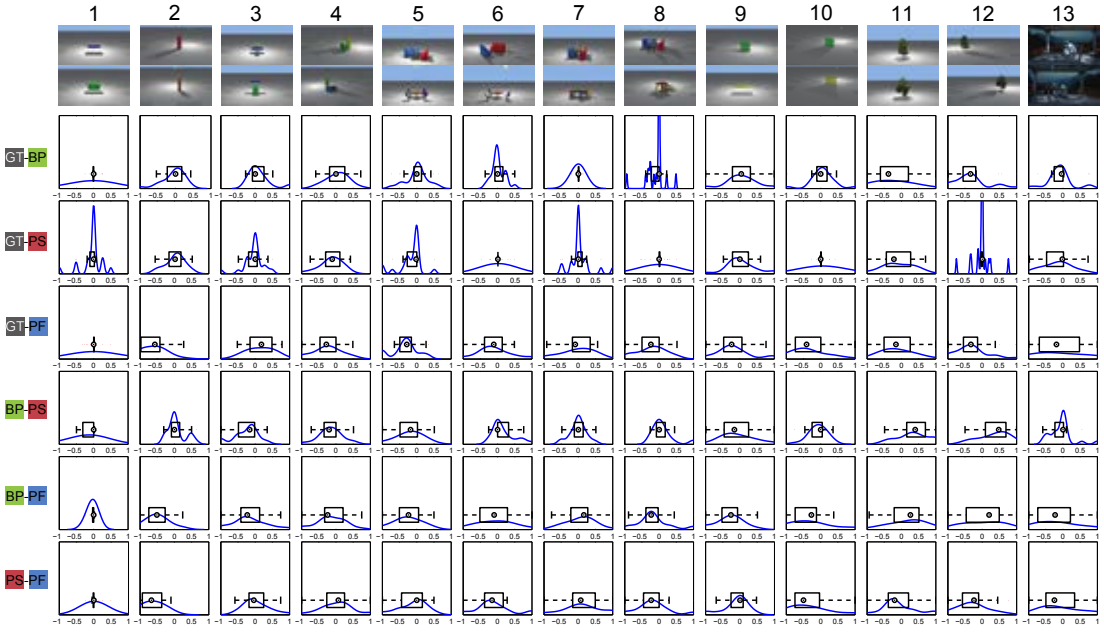

Figure A.5: Slider distribution in Block III Group B (“Which is physically more plausible?”).

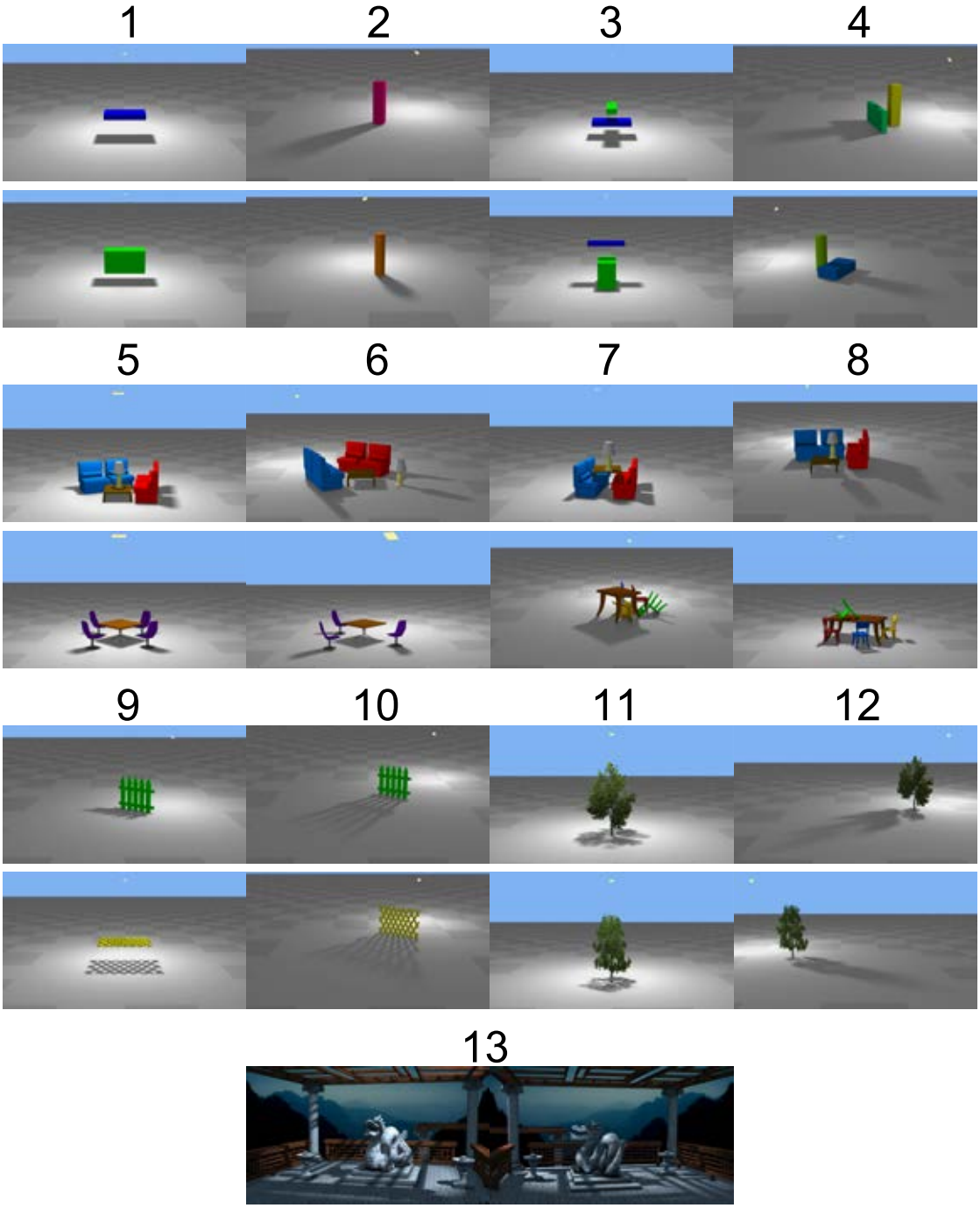

Figure A.6: Overview of all images used for the study.

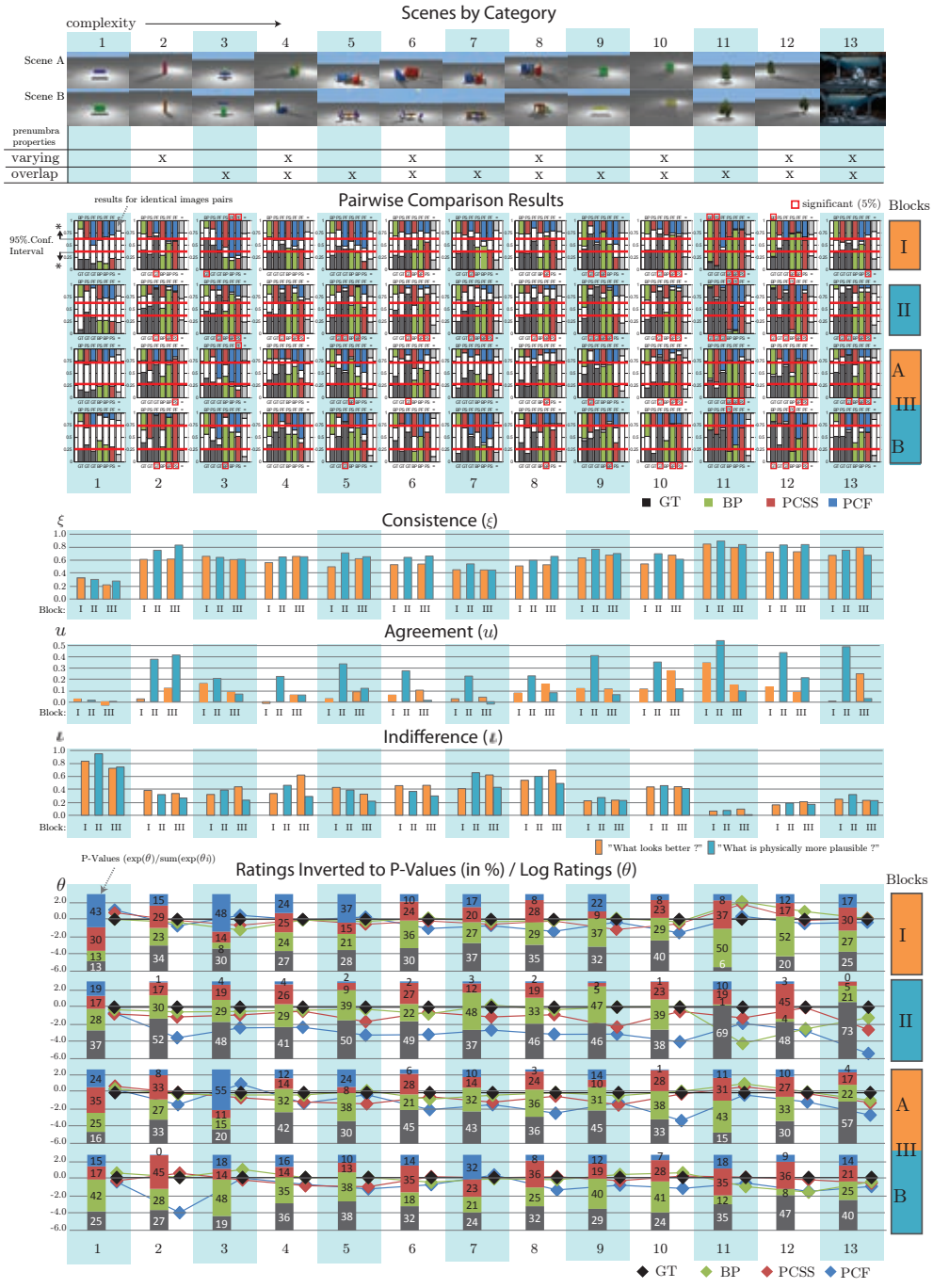

Figure A.7: Overview on all results analyzed for each of the 13 scenes separately.

Bibliography

[1] Tomas Akenine-Möller and Ulf Assarsson. Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, pages 297–306, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[2] Ulf Assarsson and Tomas Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.*, 22(3):511–520, July 2003.

[3] Ulf Assarsson, Michael Dougherty, Michael Mounier, and Tomas Akenine-Möller. An optimized soft shadow volume algorithm with real-time performance. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, HWWS '03, pages 33–40, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[4] Lionel Atty, Nicolas Holzschuch, Marc Lapierre, Jean-Marc Hasenfratz, Charles Hansen, and Francois X. Sillion. Soft shadow maps: Efficient sampling of light source visibility. *Computer Graphics Forum*, 25(4):725–741, 2006.

[5] C. James Bartleson and Franc C. Grum. *Chapter 8. In Optical Radiation Measurements*, volume 5 of *Visual Measurements*. Eds. Academic Press, 1984.

[6] J. Blinn. Me and my (fake) shadow. *IEEE Comput. Graph. Appl.*, 8(1):82–86, January 1988.

[7] Pierre Boulenguez, Boris Airieau, Mohamed-Chaker Larabi, and Daniel Meneveaux. Towards a perceptual quality metric for computer-generated images. In *Proceedings of SPIE - The International Society for Optical Engineering*, SPIE '12, 2012.

[8] Ralph A. Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4), 1952.

[9] Xiao-Hua Cai, Yun-Tao Jia, Xi Wang, Shi-Min Hu, and Ralph R. Martin. Rendering soft shadows using multilayered shadow fins. *Computer Graphics Forum*, 25(1):15–28, 2006.

[10] Eric Chan and Frédo Durand. Rendering fake soft shadows with smoothies. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 208–218, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[11] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5:51–72, January 1986.

[12] Douglas Critchlow and Michael Fligner. Paired comparison, triple comparison, and ranking experiments as generalized linear models, and their implementation on GLIM. *Psychometrika*, 56(3):517–533, September 1991.

[13] Franklin C. Crow. Shadow algorithms for computer graphics. *Computer Graphics (SIGGRAPH '77 Proceedings)*, 11:242–248, July 1977.

[14] Herbert David. *The Method of Paired Comparison*. Charles Griffin and Company, 1963.

[15] Roger R. Davidson. On extending the bradley-terry model to accommodate ties in paired comparison experiments. Technical report, FSU Statistics Report Ml69 ONR Technical Report No. 37, Department of Statistics Florida State University, 1969.

[16] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, I3D '06, pages 161–165, New York, NY, USA, 2006. ACM.

[17] Elmar Eisemann and Xavier Décoret. Plausible image based soft shadows using occlusion textures. In Rodrigo Lima Oliveira Neto, Manuel Menezes deCarceroni, editor, *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, 19 (SIBGRAPI)*, Conference Series. IEEE, IEEE Computer Society, 2006.

[18] G.T. Fechner. *Elemente der Psychophysik*. Number Bd. 2 in Elemente der Psychophysik. Breitkopf und Härtel, 1860.

[19] Randima Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[20] James A. Ferwerda. Psychophysics 101: how to run perception experiments in computer graphics. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 87:1–87:60, New York, NY, USA, 2008. ACM.

[21] Stephen Fienberg, Paul Holland, and Yvonne Bishop. *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, 1975.

[22] George A. Gescheider. *Psychophysics: The Fundamentals*. Lawrence Erlbaum, 3 edition, 5 1997.

[23] Gaël Guennebaud, Loïc Barthe, and Mathias Paulin. Realtime soft shadow mapping by backprojection. *Computer Graphics Forum*, pages 227–234, 2006.

[24] Diego Gutierrez, Francisco J. Seron, Jorge Lopez-Moreno, Maria P. Sanchez, Jorge Fandos, and Erik Reinhard. Depicting procedural caustics in single images. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 120:1–120:9, New York, NY, USA, 2008. ACM.

[25] Eric Haines. Soft planar shadows using plateaus. *Journal of Graphics Tools*, 6:2001, 2001.

[26] Adrian Jarabo, Tom Van Eyck, Veronica Sundstedt, Kavita Bala, Diego Gutierrez, and Carol O'Sullivan. Crowd Light: Evaluating the Perceived Fidelity of Illuminated Dynamic Scenes. *Computer Graphics Forum*, 31(2):565–574, 2012.

[27] Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[28] Maurice Kendall and Jean D. Gibbons. *Rank Correlation Methods*. A Charles Griffin Title, 5 edition, September 1990.

[29] Florian Kirsch and Juergen Doellner. Real-time soft shadows using a single light sample. *Journal of WSCG (Winter School on Computer Graphics*, 11:2003, 2003.

[30] Andrew Lauritzen and Michael McCool. Layered variance shadow maps. In *Proceedings of graphics interface 2008*, GI '08, pages 139–146, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[31] Patrick Ledda, Alan Chalmers, Tom Troscianko, and Helge Seetzen. Evaluation of tone mapping operators using a high dynamic range display. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 640–648, New York, NY, USA, 2005. ACM.

[32] D. Brandon Lloyd, Naga K. Govindaraju, Cory Quammen, Steven E. Molnar, and Dinesh Manocha. Logarithmic perspective shadow maps. *ACM Trans. Graph.*, 27(4):106:1–106:32, November 2008.

[33] Katerina Mania and Andrew Robinson. The effect of quality of rendering on user lighting impressions and presence in virtual environments. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, VRCAI '04, pages 200–205, New York, NY, USA, 2004. ACM.

[34] Tobias Martin and Tiow-Seng Tan. Anti-aliasing and Continuity with Trapezoidal Shadow Maps . pages 153–160.

[35] Parker, Steven, Peter Shirley, and Brian Smits. Single sample soft shadows. *Tech. Rep. UUCS-98-019*, 1998.

[36] Robert Pietsch. *Statistische Herausforderungen sozialwissenschaftlicher Studien: Bradley-Terry-Luce Modell*. Ludwig-Maximilians-University Munich.

[37] Laura Raya, Susana Mata, and Óscar D. Robles. Perceptual evaluation of illumination effects in virtual environments. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, APGV '10, pages 167–167, New York, NY, USA, 2010. ACM.

[38] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 283–291, 1987.

[39] Michael Rubinstein, Diego Gutierrez, Olga Sorkine, and Ariel Shamir. A comparative study of image retargeting. *ACM Trans. Graph.*, 29:160:1–160:10, December 2010.

[40] Mirko Sattler, Ralf Sarlette, Thomas Mücken, and Reinhard Klein. Exploitation of human shadow perception for fast shadow rendering. In *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, APGV '05, pages 131–134, New York, NY, USA, 2005. ACM, ACM.

[41] Henry Scheffe. An analysis of variance for paired comparisons. *Journal of the American Statistical Association*, 47(259):381–400, 1952.

[42] Michael Schwarz and Marc Stamminger. Bitmask soft shadows. *Computer Graphics Forum*, 26(3):515–524, 2007.

[43] Michael Schwarz and Marc Stamminger. Microquad soft shadow mapping revisited. In *Eurographics 2008 Annex to the Conference Proceedings (Short Papers)*, Eurographics 2008, pages 295–298. Eurographics, Eurographics Association, 2008.

[44] Michael Schwarz and Marc Stamminger. Quality scalability of soft shadow mapping. In *Proceedings of graphics interface 2008*, GI '08, pages 147–154, Toronto, Ont., Canada, Canada, 2008. ACM, Canadian Information Processing Society.

[45] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, and Paul Haeberli. Fast shadows and lighting effects using texture mapping. *SIGGRAPH Comput. Graph.*, 26(2):249–252, July 1992.

[46] I. Setyawan and R. L. Lagendijk. L.: Human perception of geometric distortions in images. In *Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents VI*. SPIE, 2004.

[47] Marc Stamminger and George Drettakis. Perspective shadow maps. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 557–562, New York, NY, USA, 2002. ACM.

[48] Juraj Vanek, Jan Navrátil, Adam Herout, and Pavel Zemčík. High-quality shadows with improved paraboloid mapping. In *Proceedings of the 7th international conference on Advances in visual computing - Volume Part I*, ISVC'11, pages 421–430, Berlin, Heidelberg, 2011. Springer-Verlag.

[49] Peter Vangorp, Olivier Dumont, Toon Lenaerts, and Philip Dutré. A perceptual heuristic for shadow computation in photo-realistic images. In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH '06, New York, NY, USA, 2006. ACM, ACM.

[50] Leonard Wanger. The effect of shadow quality on the perception of spatial relationships in computer generated imagery. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 39–42, New York, NY, USA, 1992. ACM.

[51] Leonard C. Wanger, James A. Ferwerda, and Donald P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Comput. Graph. Appl.*, 12:44–51, 54–58, May 1992.

[52] Lance Williams. Casting curved shadows on curved surfaces. *Computer Graphics (SIGGRAPH '78 Proceedings)*, 12(3):270–274, Aug. 1978.

[53] Michael Wimmer, Daniel Scherzer, and Werner Purgathofer. Light space perspective shadow maps. In *Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering)*, Eurographics 2004. Eurographics Association, 2004.

[54] Chris Wyman and Charles Hansen. Penumbra maps: approximate soft shadows in real-time. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 202–207, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[55] Insu Yu, Andrew Cox, Min H. Kim, Tobias Ritschel, Thorsten Grosch, Carsten Dachsbacher, and Jan Kautz. Perceptual influence of approximate visibility in indirect illumination. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, APGV '09, pages –, New York, NY, USA, 2009. ACM.

[56] Fan Zhang. *Anti-aliased shadow mapping for large-scale and dynamic scenes*. PhD thesis, 2007. AAI3302440.